Polo del Conocimiento

Pol. Con. (Edición núm. 51) Vol. 5, Especial No 1
Noviembre Especial 2020, pp. 880-899
ISSN: 2550 - 682X
DOI: 10.23857/pc.v5i1.2022

Simulación del modelo matemático de un robot móvil diferencial con control de posición sin orientación basado en ley de Lyapunov

Simulation of the mathematical model of a differential mobile robot with orientation-free position control based on Lyapunov's law

Simulação do modelo matemático de um robô móvel diferencial com controle de posição livre de orientação baseado na lei de Lyapunov

Fabian Israel Heredia-Moreno[I]
fabian.heredia@espoch.edu.ec
https://orcid.org/0000-0002-9413-7504

Michael Estefania Játiva-Brito[II]
misheljativa19@gmail.com
https://orcid.org/0000-0002-6394-2586

Cristian Geovanny Merino-Sánchez[III]
cristian_merino13@hotmail.es
https://orcid.org/0000-0003-3645-5165

Adriana Virginia Macías-Espinales[IV]
adrianita_23j@hotmail.com
https://orcid.org/0000-0002-5303-6520

Correspondencia: fabian.heredia@espoch.edu.ec

Ciencias de la Computación
Artículo de investigación

*__Recibido:__ 13 de septiembre de 2020 *__Aceptado:__ 09 de octubre de 2020 * __Publicado:__ 25 de noviembre de 2020

I. Ingeniero en Electrónica, Control y Redes Industriales, Especialista de Proyectos y Transferencia de Tecnologías, Escuela Superior Politécnica del Chimborazo, Riobamba, Ecuador.

II. Ingeniera en Electrónica Control y Redes Industriales, Docente, Centro de Capacitación Fundel, Ecuador.

III. Master Universitario en Ingeniería de Software y Sistemas Informáticos, Ingeniero en Sistemas Informáticos, Escuela Superior Politécnica del Chimborazo, Riobamba, Ecuador.

IV. Magister en Tecnología e Innovación Educativa, Ingeniera en Sistemas, Docente, Universidad Laica Eloy Alfaro de Manabí, Manta, Ecuador.

Resumen

El presente artículo trata de la simulación para situar un robot móvil diferencial en un punto de referencia sin orientación basado en leyes de Lyapunov. Para lo cual se ha considerado un ambiente de trabajo estructurado, donde la herramienta principal es el software matemático Matlab, mismo que se es el protagonista dentro de la simulación del robot móvil diferencial. Como punto de partida se ha procedido a realizar cálculos cinemáticos y dinámicos, siendo una de las partes fundamentales de esta investigación para comprender el comportamiento dentro de la simulación del robot. El manejo de los controladores en este trabajo es una parte importante debido a que los mismos se ha encargado de brindar un ajuste detallado a los cálculos realizados para la simulación del robot móvil. Los parámetros de los controladores se los ha desarrollado mediante un conjunto de funciones descritas en el entorno de Matlab.

Palabras Clave: Robot móvil; simulación; matlab; leyer de lyapunov; cálculo cinemático; cálculo dinámico; controlador.

Abstract

This article deals with the simulation to locate a differential mobile robot in a reference point based on Lyapunov laws. For that, a structured working environment has been considered, from which the main tool is the Matlab mathematical software, which is the protagonist within the simulation of the differential mobile robot. As a starting point, cinematic and dynamic calculations have been carried out, being one of the fundamental parts of this investigation to understand the behavior within the robot simulation. The management of the controllers in this work is an important part due to the fact that the monks have been responsible for providing a detailed adjustment to the calculations made for the simulation of the mobile robot. The parameters of the controllers have been developed using a set of functions described around Matlab.

Keywords: Mobile robot; simulation; matlab; lyapunov's leyer; Cinematic calculation; Dynamic calculation; Controller.

Resumo

O presente artigo trata da simulação para situar um robô móvel diferencial em um ponto de referência sem orientação baseado em leyes de Lyapunov. Para que cual seja considerado um ambiente de trabalho estruturado, faça o herramienta principal é o software matemático

Matlab, mismo que é o protagonista dentro da simulação do robô móvel diferencial. Como ponto de partida se há procedido para realizar cálculos cinemáticos e dinâmicos, siendo una de las partes fundamentales de esta investigação para comprender el comportamiento dentro de la simulación del robot. O manejo de los drivers neste trabalho é uma parte importante debido a que los mismos se encargado de brindar um ajuste detalhado e los cálculos realizados para a simulação do robô móvil. Os parâmetros dos drivers são desarrollados mediante um conjunto de funções efetuadas no entorno de Matlab.

Palavras-chave: Robô móvel; simulação; matlab; leyer de lyapunov; cálculo cinemático; cálculo dinâmico; controlador.

Introducción

Los robots móviles a menudo se hallan en escenarios en las que tienen que trasladarse de un punto a otro enfrentando los inconvenientes del entorno como son los obstáculos y las limitaciones de su capacidad. (Bruce & Veloso, 2002).

A menudo los métodos de control de movimiento requieren altos costes computacionales y errores numéricos que lo hacen inadecuados para aplicaciones en tiempo real. (Frazzoli, Dahleh, & Feron, 2005).

Los métodos de planificación para rutas, se desarrollan continuamente para problemas cada vez más desafiantes y con costos computacionales más bajos, todas las investigaciones apuntan a lograr mecanismos móviles autónomos, este objetivo es posible lograrlo con el uso de métodos reactivos que permiten al robot interactuar con el entorno de trabajo.

El problema básico de un robot móvil con ruedas está enfocado en encontrar el camino de un punto inicial a un punto final, evadiendo obstáculos que se presentan en el ambiente de trabajo.

La propuesta contempla en un algoritmo matemático para un robot móvil diferencial desde un punto de partida hacia un punto de llegada, en un ambiente estructurado con control de posición sin orientación.

Diseño del sistema

Diseño de sistemas de control no lineal: usando Lyapunov

La teoría de estabilidad de Lyapunov es una herramienta estándar y una de las más importantes en el análisis de sistemas no lineales, además permite analizar la estabilidad de un

sistema de control y puede ser aplicada como una estrategia para el diseño de controladores, el objetivo de este método es encontrar una función candidata de Lyapunov, cumpliendo con ciertas condiciones asegurando así la estabilidad de los puntos de equilibrio.

Sistema estable

La presente investigación es de tipo cuantitativa, debido a que se fundamenta en una valoración numérica del software, tanto desde la perspectiva de los usuarios, así como una valoración técnica de las características del producto con base en los resultados brindados por dos herramientas de testeo (WebPagetest y MatchMetrics), de las cuales se tomaron en cuenta únicamente los parámetros determinados por la norma ISO/IEC 25010, todo esto como un estudio de campo. Posteriormente, se utilizó la investigación descriptiva para detallar los hallazgos (potencialidades y deficiencias) del software y las conclusiones del estudio.

La función principal de Lyapunov V(x) es una función de estados x que cumple con las siguientes condiciones.

1. V(x) es continua y sus derivadas parciales son continuas

Función definida positiva

2. $V(x) = 0$ para $x = 0$

3. $V(x) > 0$ para $x \neq 0$

Función semi-definida negativa

4. $(dV(x))/dt=0$ para $x=0$

5. $(dV(x))/dt \leq 0$ para $x \neq 0$

Las condiciones son fundamentales para que el sistema de control sea considerado un sistema estable. De esta manera se utilizó la teoría de Lyapunov para realizar algoritmos de control, por lo tanto, los estados x serán los errores, y los puntos de equilibrio que van serán el origen,

es decir el punto cero, esto debido a que el objetivo del control es llegar a que el error del mismo vaya hacia cero de esta manera las condiciones para que el sistema sea estable son las siguientes:

1. La función candidata de Lyapunov y sus derivadas parciales debe ser continua.
2. La función candidata de Lyapunov debe ser igual a cero solo para x igual a cero
3. La función debe ser mayor a cero solo para valores diferentes de cero
4. La derivada de la función candidata de Lyapunov debe ser igual a cero para x igual a cero.
5. La derivada de la función candidata de Lyapunov debe ser menor igual a cero para x diferente de cero.
6. V(x) es continua y sus derivadas parciales son continuas
7. Función definida positiva
8. V(x) = 0 para x = 0
9. V(x) > 0 para x ≠0
10. Función semi-definida negativa
11. $\frac{dV(x)}{dt} = 0$ para $x = 0$
12. $\frac{dV(x)}{dt} \leq 0 \; para \; x \neq 0$

Metodología

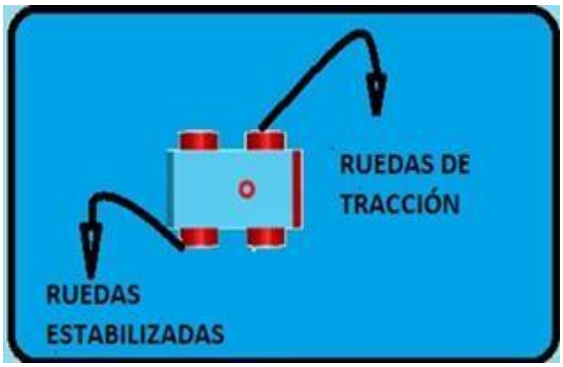
Diseño de sistemas de control no lineal basados en el modelo

Las consideraciones para el diseño del controlador no lineal basado en el modelo e implementado la teoría de Lyapunov son las siguientes:

- Dado un sistema físico (modelo matemático y sus restricciones) y su comportamiento deseado, el objetivo obtener una ley de control de modo que el sistema realimentado se comporte como se desea.
- Si las tareas del sistema de control involucran gran cantidad de las variables o alta velocidad, muy probablemente deba recurrirse a un diseño de control no lineal al fin de alcanzar las especificaciones.

Los robos móviles diferencial cuenta con dos pares de ruedas: esto quiere decir dos ruedas de tracción que tienen acoplados dos motores DC, y dos ruedas estabilizadas que mantienen el balance del vehicúlalo (figura 1).

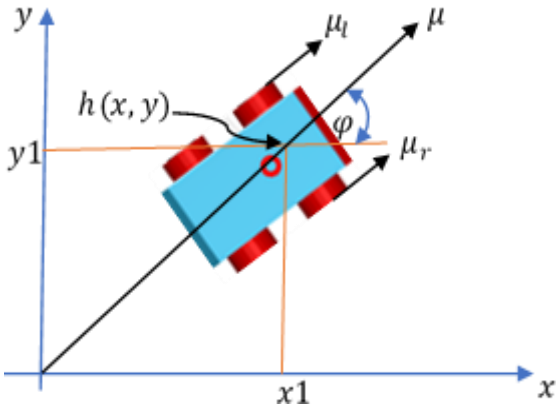
Figura 1-3: robot móvil diferencial.



La traslación y rotación de este tipo de plataformas diferenciales se determina por el movimiento independiente de cada una de las ruedas de tracción.

Para proceder al cálculo cinemático se trazó el sistema de referencia, en este caso en el palo x, y. A continuación, se inserta el robot móvil diferencial en una posición diferente a la del origen x1, y1 figura 2.

Figura 2-3: calculo cinemático.



Datos:

φ orientación

μ_(l)velocidad lineal de la llanta derecha

μ_(r)velocidad lineal de la llanta izquierda

h(x,y) posición del robot en el punto de control en este caso es en el centro del eje de las ruedas.

Sabiendo que el punto de control es el punto de interés para conocer la posición.

El sistema total del robot tendrá una velocidad lineal que permita al robot moverse hacia delante y hacia atrás.

La velocidad angular μ que permite que el robot gire en sentido horario y en sentido antihorario.

Modelo Cinemático de un robot móvil diferencial

Como se puede observar en la figura 2-3 el robot se ha desplazado:

$$h_x = x_1$$
$$h_y = y_1$$

Si se deriva la posición se obtiene la velocidad, Por consiguiente, la velocidad se va a descomponer en el eje (x,y) por lo tanto se tendrá que la velocidad en el punto x será igual a la velocidad lineal.

$$\dot{h_x} = \mu \cos\varphi$$

Ecuación 1

$$\dot{h_y} = \mu \sin\varphi$$

Ecuación 2

y la derivada de la orientación da como resultado la velocidad angular ω.

φ ω

Ecuación 3

La velocidad total del sistema será el promedio de la velocidad de la llanta derecha μ_r mas promedio de la velocidad de la llanta izquierda.

μ= (μ_r+μ_l) /2

Ecuación 4

La velocidad angular se relaciona con la velocidad lineal de la llanta derecha menos la velocidad angular de la llanta izquierda por una distancia d.

d es la distancia entre las ruedas

$$\omega = \frac{\mu_r - \mu_l}{d}$$

Ecuación 5

Una vez que se tenga planteadas las ecuaciones, se puede remplazar la ecuación 3 en la ecuación 1 y en la ecuación 2

$$\dot{h_x} = \frac{\mu_r + \mu_l}{2}\cos\varphi$$

Ecuación 6

Diseño de controladores

Suponiendo que al robot móvil se le asigna una trayectoria de movimiento h(t) al punto de interés en términos de h ̇(t) y sus condiciones iniciales de posición y orientación, determinando un conjunto de velocidades q (̇ t) que reduzca la trayectoria dada. Se considera un sistema que tenga el mismo número de ecuaciones que incógnitas que m=n, las velocidades de los actuadores se pueden obtener mediante la inversión de la matriz jacobiana.

$$\dot{q}(t) = J\big(q(t)\big)^{-1}\dot{h}(t)$$

Ecuación 10

$\dot{q}(t)$velocidades de los actuadores

$J\big(q(t)\big)^{-1}$ matriz inversa Jacobiano

$\dot{h}(t)$ velocidades en el punto de interés

Sin embargo, las velocidades $\dot{q}(k)$ en tiempo discreto no coinciden con la $\dot{q}(t)$ velocidades en tiempo continuo, este inconveniente se puede supera a un esquema de solución que tenga en cuenta el **ERROR** entre la posición y orientación del punto de interés deseado y real.

$$h_e = h_d - h$$

Ecuación 11

h_e error

h_d velocidad de la trayectoria deseada

h punto real

Sí se aplica la deriva al error:

$$\dot{h}_e = \dot{h}_d - \dot{h}$$

Despejando $\dot{h}$

$$\dot{h} = \dot{h}_d - \dot{h}_e$$

Ecuación 12

Las velocidades de los actuadores son:

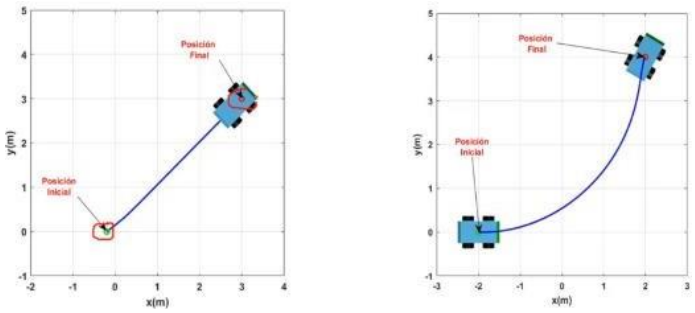
$$\dot{q}(t) = J\big(q(t)\big)^{-1}(\dot{h}_d - \dot{h}_e)$$

Ecuación 13

Control de posición sin orientación

El control de posición consiste en ubicar al robot en un punto de referencia deseado con o sin una orientación deseada figura 3-3

Figura 3-3: calculo cinemático.



Partiendo de la ecuación 13, en este caso h_d la trayectoria deseada será constante porque solo se va a posicionar en un punto al robot, por lo tanto, la constante a lo largo del tiempo es la misma.

De esta manera la derivada de la contante es $\dot{h}_d = 0$

Por lo tanto, se obtiene que la velocidad de los actuadores:

$$\dot{q}(t) = J(q(t))^{-1}(-\dot{h}_e)$$

Ecuación 14

Estabilidad de Lyapunov

El objetivo de aplicar la estabilidad de Lyapunov es asegurar que el controlador sea asintóticamente estable.

Para el análisis de estabilidad se asume el seguimiento perfecto de velocidades.

$$\dot{q}_{ref} = \dot{q}$$

Ecuación 15

Donde la velocidad de referencia (controlador) serám iguales a las velocidades de lectura que tienen el robot diferencial.

Considerando la función candidata de Lyapunov.

$$V(h_e) = \frac{h_e\ Th_e}{2}$$

Ecuación 16

$$\dot{V} = h_e\ T\dot{h}_e$$

Ecuación 17

$$\dot{h}_e = -kh_e$$

Ecuación 18

La función candidata de Lyapunov debe cumplir los términos asintóticamente estables antes mencionados.

1. $V(h_e)$ la función candidata debe ser continua si cumple.
2. $V(h_e)$ Función definida es positiva cumple con la condición
3. Para el análisis de la derivada se debe asegurar que la derivada sea definida negativa, en este caso no se utilizara la derivada del error en algoritmo de control (ecuación 17) de esta manera se utilizara la ecuación 18 en la cual se asume que $\dot{h}_e = -kh_e$ de esta manera se obtiene la ecuación:

$$\dot{V} = -h_e\, Tkhe < 0$$

Donde, k es una matriz definida positiva (generalmente diagonal) para que el sistema sea asintóticamente estable se obtiene:

$$\dot{q}_{ref} = J(q(t))^{-1}(kh_e)$$

Ecuación 19

En las cuales las velocidades de referencia que se aplicará a los actuadores son iguales a la inversa de la matriz jacobiana por una matriz k diagonal definida positiva por el error.

Con esto se desarrolla el algoritmo de control a implementarse en Matlab.

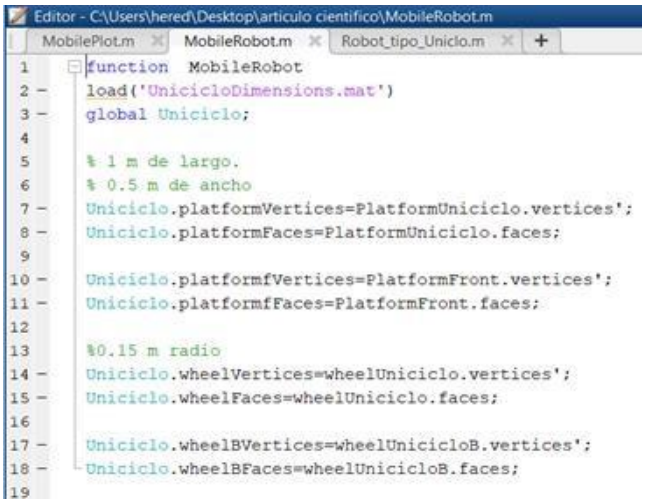
Emulación del robot móvil diferencial

Para proceder a realizar la simulación del robot primero se creó dos archivos.

- MobilPlot.m
- MobileRobot.m

En el archivo MobilRobot.m se almacenarán todos los parámetros del robot en el cual se encuentran las dimensione del mismo, tamaño de las llantas, colores etc.

Figura 4-3: archivo de Matlab MobilPlto.m



Y el archivo MobilPlot.m permitirá dibujar al robot en la posición de x y y con una orientación phi, para esto se utilizó matrices de traslación y de rotación figura 5-3

Figura 5-3: matrices de rotación

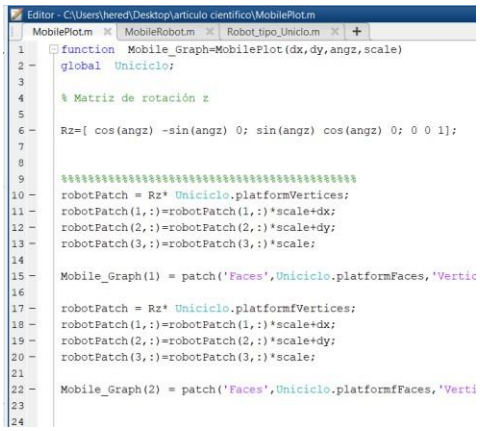
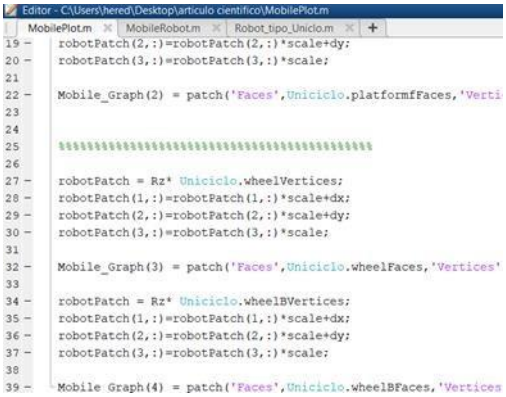


Figura 6-3: matrices de traslación



Implementación del Controlador en Matlab

Para la implementación del controlador en Matlab se calculó el error con el siguiente código:

```
>> hxe(k) = hxd - hx(k);
>> hye(k) = hyd - hy(k);
>> he = [hxe(k) hye(k)]';
```

Establecida en la ecuación 11, donde hxe y hye son las posiciones son las posiciones reales del robot.

Lugo se generó la matriz jacobiana para la simulación:

```
J11 = cos(phi(k));
```

```
J12 = -a*sin(phi(k));

J21 = sin(phi(k));
J22 = a*cos(phi(k));

J= [J11 J12;... J21 J22];
```

El parámetro de control a utilizar es matriz diagonal definida positiva:

```
K = [0.3 0; ..0 0.3];
```

Luego se insertó la ley de control hallada en la ecuación 19.

```
qpref = pinv(J)*(K*tanh(he));
```

Para obtener la inversa de la matriz jacobiana se utilizó la función pinv propia de Matlab, obteniendo las velocidades de referencia que se aplicara al robot diferencial.

Se separó las acciones de control en velocidad lineal y angular.

```
>> uRef(k) = qpref(1);% velocida lineal
>> wRef(k) = qpref(2);%velocidad angúlar
```

Aplicar acciones de control al robot móvil simulado

Se calculó la orientación real del siguiente instante de muestreo utilizando las acciones de control que se obtuvo en el modelo.

1. Robot simulado (modelo cinemático)

```
>> phi(k+1)=phi(k)+ts*wRef(k);% orientación real en radianes
>> x1p=uRef(k)*cos(phi(k+1)); % velocidad real en metros / segundos
>> y1p=uRef(k)*sin(phi(k+1)); % velocidad real en metros / segundos
```

Con el método de Euler se halló las posiciones en los siguientes instantes de muestreo del eje central de las ruedas, considerando que no es el punto de interés o el punto de control que se

calcula para dibujar el robot en el cual el robot debe ingresar la posición central en el eje de las ruedas

2. Integral numérica (método de Euler)

```
>> x1(k+1)=x1(k)+ts*x1p; % posición real del centro (eje x) en metros (m)
>> y1(k+1)=y1(k)+ts*y1p; % posición real del centro (eje y) en metros (m)
```

Se calculó el punto de interés donde hx y hy nuestros puntos de control o interés.

3. modelo geométrico

```
>>hx(k+1)=x1(k+1)+a*cos(phi(k+1));   % posición de interés real (eje x) en metros (m)
 >>hy(k+1)=y1(k+1)+a*sin(phi(k+1));   % posición de interés real (eje y) en metros (m)
```

Resultados, discusión y análisis

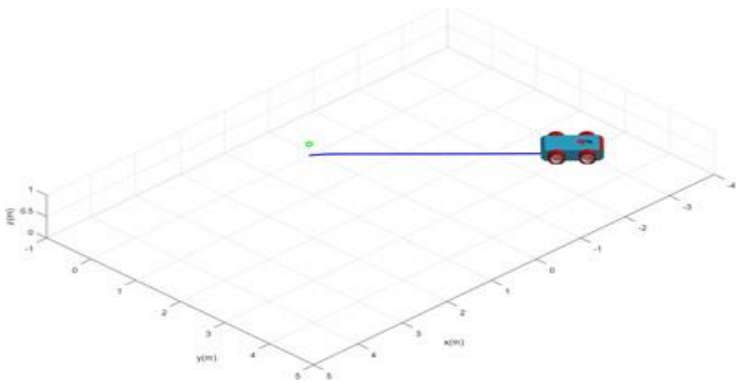
Una vez terminado el programa de Matlab se procedió a modificar los valores para obtener los resultados del modelo matemático implementado.

Validación del modelo

Para la validación del modelo se ejecutó la programación en el software de Matlab para esto se ingresó los parámetros de su posición inicial.

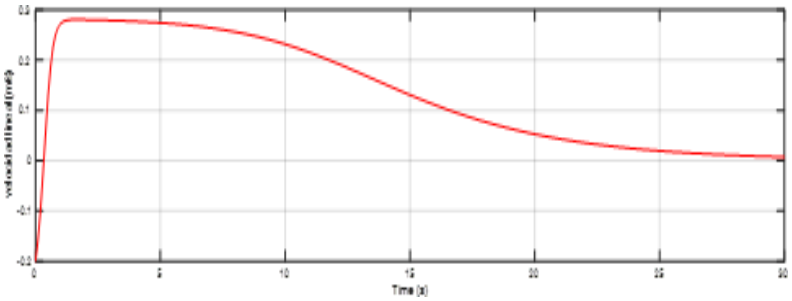
```
>> hxd = -3;
>> hyd = 3;
```

Figura 7-4: simulación del robot móvil desde la posición inicial a su posición final sin considerar su orientación.



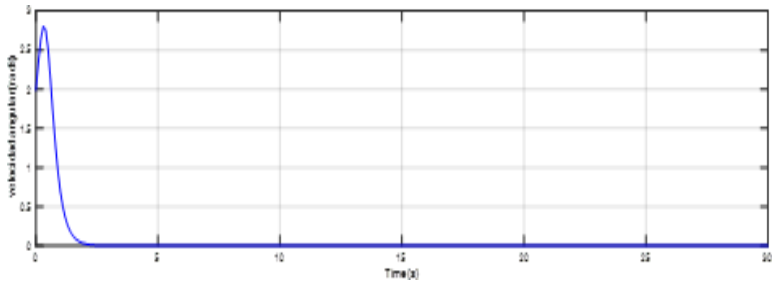
Al realizar la simulación se observó que el robot se aproximan al punto desea (3;-3).

Figura 8-4: velocidad lineal



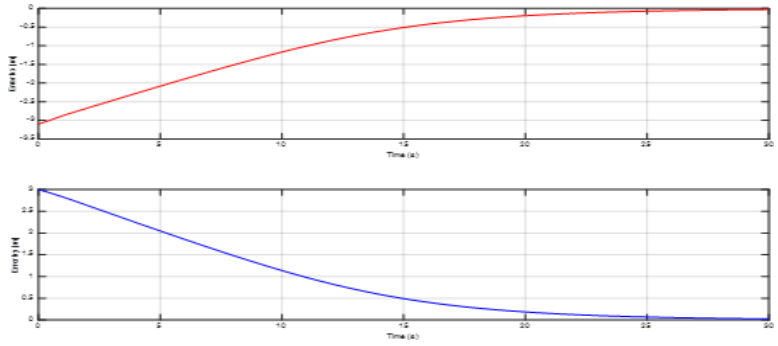
Como se puede observar en la figura la velocidad lineal es de 0.41 m/s luego va decreciendo hasta llegar a 0 m/s

Figura 9-4: velocidad angular



La velocidad angular se dispara alcanzando los 4 rd/seg y luego se estabiliza.

Figura 10 -4: errores h_x y h_y



Como se puede evidenciar en la figura 3-4 los errores convergen a cero.

Realizando el cambio del parámetro de nuestras condiciones iniciales.

>> hxd = 4.5;

>> hyd = 2.8:

Figura 11-4: simulación del robot móvil desde la posición inicial a su posición final (4.5:2.8) sin considerar su orientación.

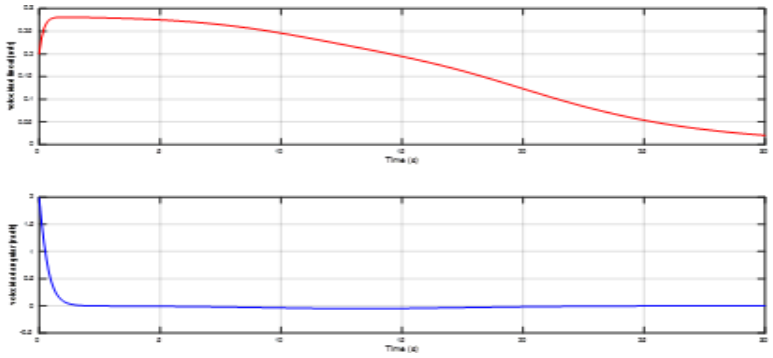
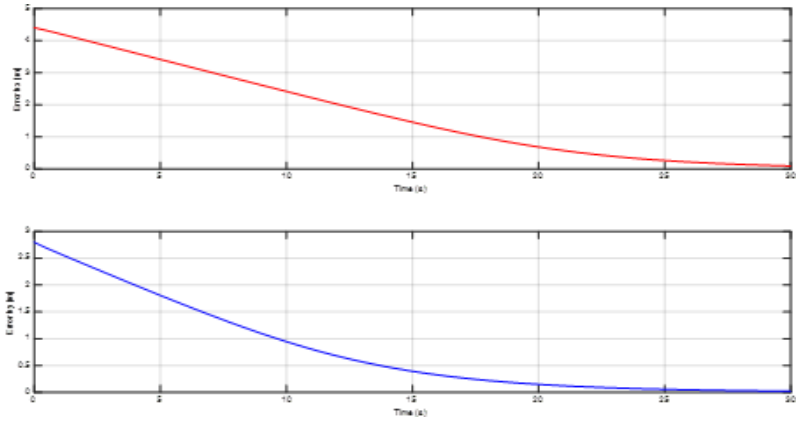


Figura 13 -4: errores h_x y h_y



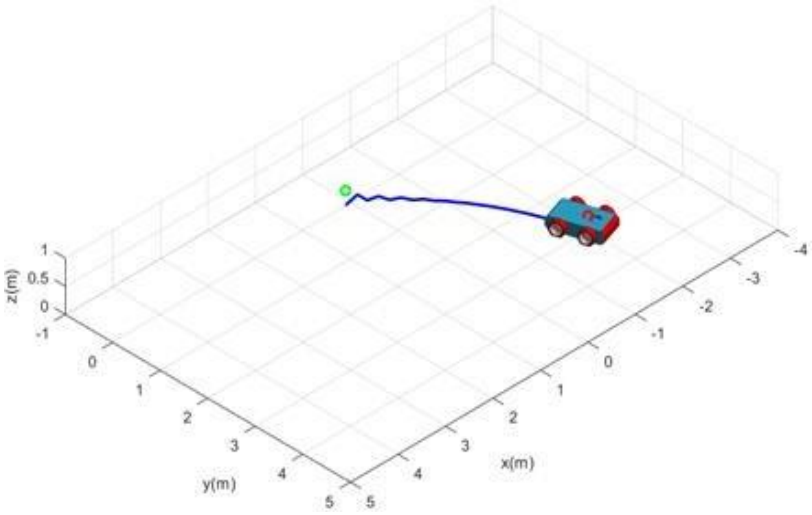
Como se puede observar en la figura 11-4 el robot se dirige a la posición deseada y su error tienden a cero con esto se logró verificar que el controlador está funcionando correctamente.

Pruebas de comportamiento

Las pruebas para verificar el comportamiento del controlador se las detalla a continuación, para lo cual es importante mencionar que el valor de k correspondiente a una constante fue tomando distintos valores. Como se mencionó en los cálculos, la constante debería ser positiva de esta manera se modificó para obtener un control.

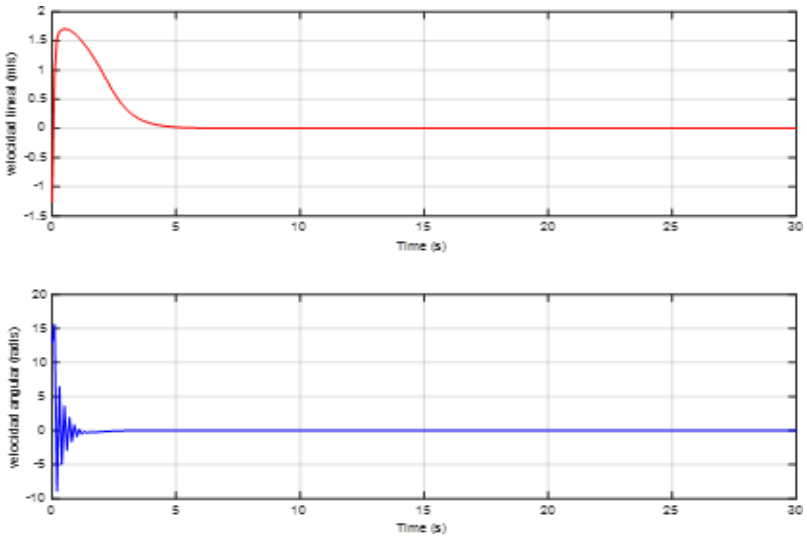
K = [1.3 0; 0 1.3];

Figura 3-8: simulación del robot móvil



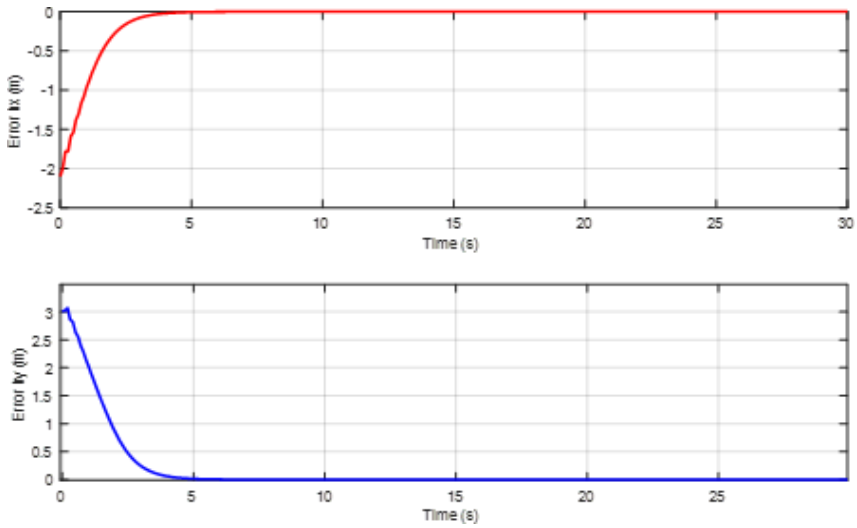
Como podemos observar en la figura el robot se traslada bruscamente

Figura 14-4: acciones de control



En las acciones de control se puede observar que existe un cambio brusco en su velocidad angular asta estabilizarse.

Figura 14-4: errores



Se puede observar que los errores decrecen más rápido a cero, pero sin embargo existen variaciones, hay que mencionar que le robot no va alcanzar estas velocidades mayores a 0.5.

Conclusiones

- Para establecer las pruebas reales no sobrepasar 1 la constante K por movimientos bruscos del robot.
- El cálculo cinemático encontrado es eficiente par que el robot móvil diferencial con control deposición sin orientación llegue al punto deseado
- El controlador implementado ha permitido que los errores se establezcan en cero dando como resultado la ubicación deseada con exactitud de 0.001.
- El modelo cinemático considera las velocidades, posición del robot móvil, sin tomar en cuenta los efectos de las fuerzas e inercias productos de la masa del robot, este modelo cinemático sirve para la implementación de un esquema de control de seguimiento de caminos
- El controlador cinemático propuesto para el robot móvil diferencial tiene como variable de control la velocidad lineal y angular del robot móvil, este controlador garantiza la estabilidad mediante el criterio de Lyapunov, demostrando que los errores de posición convergen a cero cuando el tiempo tiende al infinito.

Referencias

1. Adams. M.D, "Sensor Modelling, Design and Data Processing for Autonomous Navigation.", World Scientific Publishing, Series in Robotics and Intelligent Systems. Singapore, 1999.
2. Agre, Philip E. y Chapman David. " What are plans for?", Robotics and Autonomous Systems. No. 6 (1990); p. 17 – 34.
3. IEEE Robotics and Automatization Soviet 2001.
4. Akbarally, Huzefa y Kleeman, Lindsay. "A sonar sensor for accurate 3D target localisation and classification" IEEE International conference on robotics and automation. 1995.
5. Mataric, Maja. "A distributed model for mobile robot environment-learning and navigation."Cambridge, MA,1990.
6. Parker, L. "Current State of the Art in Dist Distributed Autonomous Mobile Robotic" ,Distributed Autonomous Robotic Systems. Tokyo. Vol 4, (2000); p. 3-12.
7. Yamasaki, H. "Intelligent Sensing Technology.", Journal of the Japan Society of Precision Engineering. Vol. 55, No. 9; 1989.
8. [8] Everett, H.R. y Peters, A K. " Sensors for MobileRobots" , Wellesley, MA,1995.
9. Feng, L, Borenstein, J y Everett, H.R. " Where am I? " Sensors and methods for mobile robot positioning. Universidad de Michigan. 1996.
10. Adams. M.D. "Sensor Modelling, Design and Data Processing for Autonomous Navigation." World cientific publishing , Series in Robotics and Intelligent Systems. Singapore. Vol. 13. 1999.
11. [mcbtec.com. "Funcionamiento_Encoder"< www.mcbtec.com/Funcionamiento_Encoder.pdf >. superrobotica.com, " sensor brújula digital "<http://www.superrobotica.com/S320160.htm >
12. Borges, G, Nogueira, A y G. S. Deep, "Characterization of a Trajectory Recognition Optical Sensor for an Automated Guided Vehicle.", IEEE Transactions on Instrumentation and measurement, Vol.49, Nº. 4, pp. 813 – 819. 2000.
13. Malik R. y Yu H., "The Infrared Detector Ring: Obstacle Detection for an Autonomous Mobile Robot." IEEE 35th Midwest Symposium on Circuit and Systems,1992.

14. Flynn A., "Combining sonar and infrared sensors for mobile robot navigation.", The International Journal of Robotics Research, Vol. 7, Nº. 6, pp. 5 – 14. 1988.
15. Sabatini A, Genovese V, Guglielmelli E, Mantuano
16. A, Ratti G, y Dario P. "A low-cost, composite sensor array combining ultrasonic and infrared proximity sensors.", Procedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 120 -126. Pittsburgh, 1995.
17. Novotny P y Ferrier N, " Using infrared sensors and the Phong illumination model to measure distances." Procedings of the IEEE