AUTOMATIZACIÓN Y ROBÓTICA
ESCUELA DE INGENIERÍA DE ANTIOQUIA

UNIVERSIDAD EIA
Ser, Saber y Servir

Asignatura: Robótica Industrial- **Docente:** M.Sc. Juan Camilo Tejada Orjuela
Programa: Ing. Mecatrónica **Final(30 %)**
04 de diciembre de 2020 **Nombre:** SANTIAGO GARCÍA ARANGO

Por favor en su hoja de examen poner la siguiente **DECLARACIÓN DE HONESTIDAD ACADÉMICA** firmando al final de la misma:

Doy mi palabra ante la Universidad y la sociedad que demanda ciudadanos comprometidos con un actuar correcto, que la presente evaluación fue desarrollada con completa honestidad y responsabilidad, atendiendo a la misión institucional de formar seres íntegros y líderes que ayuden a construir una mejor sociedad.

A continuación, encontrará el Final de la asignatura Robótica Industrial. Para el desarrollo del examen podrá sacar lápiz, lapicero, borrador, calculadora. La modalidad del examen es a distancia y es de carácter **INDIVIDUAL**. Recuerde que el examen tiene una duración de 2 horas, no obstante, la entrega del examen se cerrará en un lapso de 2.5 horas, lo que le tiempo adicional en caso de que tenga algún inconveniente con la entrega de la prueba. No obstante, tenga en cuenta que la carga sobre la plataforma aumentará en la medida en que se acerca la hora de cierre de la prueba. Es importante que en todo momento deje expresado el procedimiento que está realizando.

Lean bien todas las preguntas y verifiquen los valores proporcionados. En caso de faltar información, haga las suposiciones necesarias e indíquelas. Recuerde que este cumple como una instancia de evaluación del proceso de aprendizaje por lo que se le motiva a asumirlo de una manera responsable. Cualquier evidencia de fraude será calificada con una calificación de 0.0 y se realizará el debido proceso. **¡Éxitos!**

La empresa SAMCO Ingeniería lo ha contactado con la intensión de desarrollar una plataforma móvil de 7 ruedas. La idea es que este robot sea una plataforma omnidireccional. Como insumos de trabajo le suministran el diagrama del robot mostrado en la figura 1. Adicionalmente le informan que las ruedas suecas disponibles son de dos tipos, radio de 30mm con rodillos a 35° y de 20mm de radio con rodillos paralelos. Tenga presente que las ruedas 1, 3, 4 y 6 son suecas de rodillos inclinados y las 2, 5 y 7 son suecas de rodillos paralelos. Tenga en cuenta que las longitudes $L_1, L_2, L_3, L_4, L_5, L_6$ son iguales y miden 250mm. El chasis del robot tiene una forma geométrica de hexágono regular.

1. (2.0 puntos) Obtenga el modelo cinemático de postura y de velocidad en las ruedas aplicando el método de Canudas. Tenga presente el planteamiento de la tabla de parámetros, las matrices J y C, la selección de la matriz Σ y la definición adecuada de las ecuaciones para la generación de los modelos.

2. (2.0 puntos) Le han solicitado estructurar un sistema de navegación autónoma a partir de un autómata híbrido, para esto le han solicitado lo siguiente:

   (a) (0.6 puntos) Determine la posición de 6 sensores de proximidad que permitan una navegación mínima del robot en el plano XY. Dibújelos sobre el robot y especifique sus coordenadas, justificando su elección.

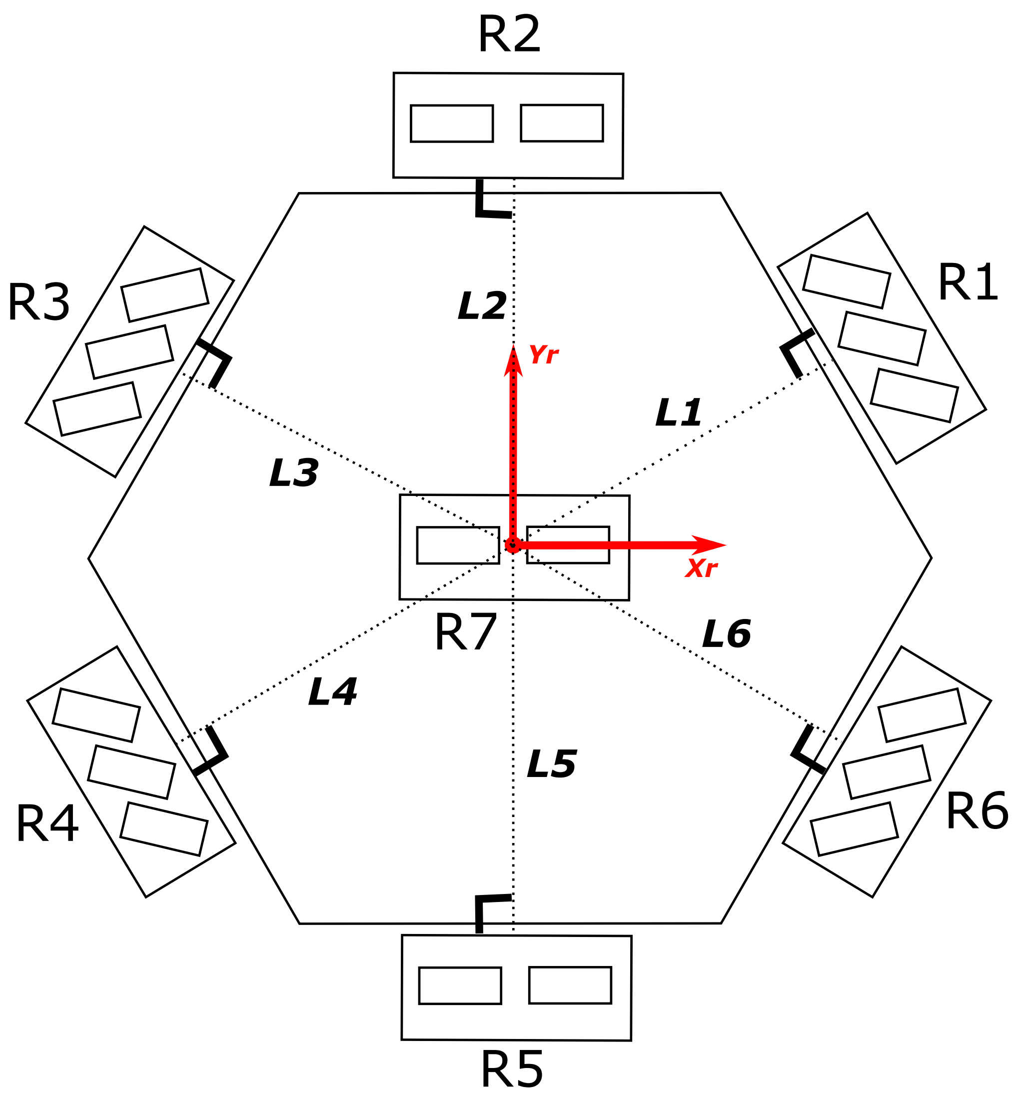


Figura 1: Robot móvil a diseñar.

(b) (1.4 puntos) Determine ecuaciones de control basados en sus sensores para determinar acciones para las tareas de:

- Evadir obstáculos.
- Seguir pared.

3. (0.5 puntos) En clase hemos planteado un modelo cinemático modificado para el robot uniciclo de la figura 2a) basado en el punto verde A de la misma, en la figura 2b), se ha propuesto calcular un modelo cinemático de postura para el punto verde B rotando 90^o el punto A.

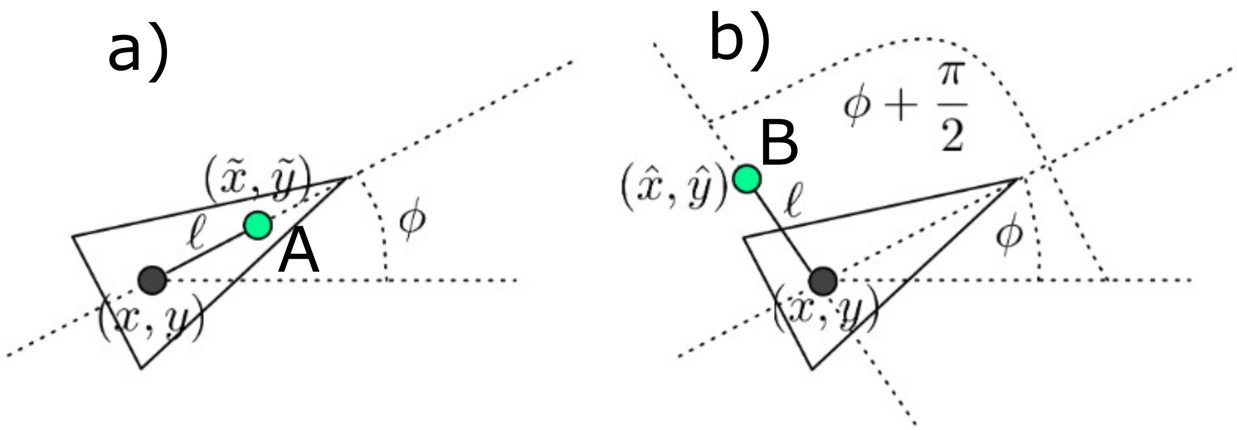


Figura 2: Modelo cinemático propuesto.

Desde su criterio de Ingeniero experto en robots móviles determine si este cambio es posible, plantee y justifique su respuesta.

4. (0.5 puntos) Conteste las siguientes preguntas considerando el libro ”Yo, robot”:

   (a) (0.25 puntos) ¿Qué quiso decir Byerley con la frase: *”Quizá habría que decir: ¡qué maravilloso! Piense que en todos los tiempos los conflictos han sido evitables. ¡Sólo las Máquinas, a partir de ahora serán inevitables!?”*

   (b) (0.25 puntos) Según la Doctora Calvin, ¿Cuál es la única razón que existiría para que un robot quebrante la primera ley y le pegue a un ser humano?

SOLUCIÓN FINAL ROBÓTICA

Santiago García Arango

SOLUCIÓN PUNTO 4

a)

Esta frase es una de las últimas mencionadas en el capítulo final de “Yo Robot”, donde se plantea el argumento final de la charla que habían tenido Stephen Byerley, con la doctora Susan Calvin. Esta frase Hace referencia a una reflexión final sobre el descubrimiento de que la razón por la que las “Máquinas” (las que manejaban la economía de los sectores principales), se habían negado a explicar el porqué tenían ciertas “falencias” en los últimos tiempos, era porque realmente las máquinas eran las que entendían el trasfondo de la humanidad, teniendo entonces una nueva primera ley que buscaa “Proteger a la humanidad misma”, a través de manejar la economía y evitar que la sociedad humanitaria las destruya.

Teniendo esto como punto de partida, entonces es un argumento muy interesante que explica que el hecho de que las máquinas tengan este nuevo poder y manejo de todo, realmente NO es algo horrible, sino que es algo positivo para la humanidad, donde la mayoría de las guerras, conflictos y enfrentamientos entre humanos, serán reducidos gracias al establecimiento de estas máquinas en la sociedad misma. También es una forma de mostrarle al lector de que tarde o temprano, las máquinas harán parte de la sociedad de forma imparable, pero que su gran objetivo es evitar que esta colapse (al menos desde el punto de vista del autor). Finalmente, luego de esta reflexión final, Susan deja un mensaje diciendo que ella vio las máquinas desde un comienzo, y ahora son ellas las únicas que se interponen entre incluso el fin de la humanidad.

b)

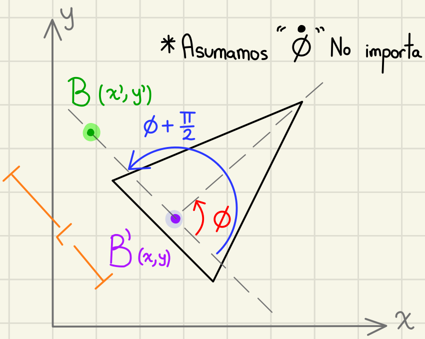
Esta pregunta se puede responder, remontándonos al capítulo de “La Prueba”. El contexto de este capítulo, es que el futuro “Organizador Mundial” (Stephen Byerley), se encontraba en su campaña política para ser alcalde. Sus competidores y enemigos tenían la creencia de que él era un robot. En resumidas cuentas, la prueba “oficial” de que él no era un robot, es que le pegó a un espectador… (supuestamente violando la primera ley)…

Sin embargo, para dar respuesta a la pregunta, al final Susan se da cuenta de que esto fue una jugada maestra de Stephen, pues es ahí donde ella dice explícitamente que:
“Sólo hay un caso en el que un robot puede pegar a un ser humano sin quebrantar la primera ley… cuando el ser humano que se golpea, es un robot”.

Teniendo en cuenta esto, realmente se puede analizar que NO existe ninguna forma de violar la primera ley de la robótica, y que lo mencionado por la doctora, es una prueba de que aunque en ese caso él haya pegado a un “humano”, realmente el elegido alcalde era un robot y aunque no se dice explícitamente esto, es una conclusión racional para el capítulo.

SOLUCIÓN PUNTO 3

*Asumamos "$\dot{\phi}$" No importa!



Posiciones Punto B.

Posiciones Punto B'.

$$x' = x + l\cos(\phi + \tfrac{\pi}{2}) = x - l\sin(\phi)$$
$$y' = y + l\sin(\phi + \tfrac{\pi}{2}) = y + l\cos(\phi)$$

$$\dot{x}' = \dot{x} - l\cos(\phi)\cdot\dot{\phi} = v\cdot c(\phi) - l\cdot c(\phi)\cdot\omega$$
$$\dot{y}' = \dot{y} - l\sin(\phi)\cdot\dot{\phi} = v\cdot s(\phi) - l\cdot s(\phi)\cdot\omega$$

$$\dot{\phi} = \omega$$

Asumámoslos como "esfuerzos de control"!

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} c(\phi) & -c(\phi) \\ s(\phi) & -s(\phi) \end{bmatrix} \cdot \begin{bmatrix} v \\ l\omega \end{bmatrix} = \begin{bmatrix} c(\phi) & -c(\phi) \\ s(\phi) & -s(\phi) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Si quisiéramos encontrar relación entre $(\dot{x}'$ y $\dot{y}')$ y $(v$ y $\omega)$, hay que resolver sistema invirtiendo matrices y despejando...

* OJO $\longrightarrow$ $\underbrace{\begin{bmatrix} c(\phi) & -c(\phi) \\ s(\phi) & -s(\phi) \end{bmatrix}}_{\heartsuit}^{-1} \longrightarrow$ NO EXISTE, porque $\boxed{\det(\heartsuit) = -c(\phi)\cdot s(\phi) + c(\phi)\cdot s(\phi) = 0}$

Desde mi criterio, este cambio NO ES POSIBLE, pues las condiciones geométricas hacen que se genere una combinación singular de senos y cosenos, generando que NO SE LOGRE encontrar una relación matemática única y valida para (V y W), con (Vx' y Vy').

Esto implica que esta cambio NO es posible y NO se debe realizar en el modelo del robot.

SOLUCIÓN PUNTO 1

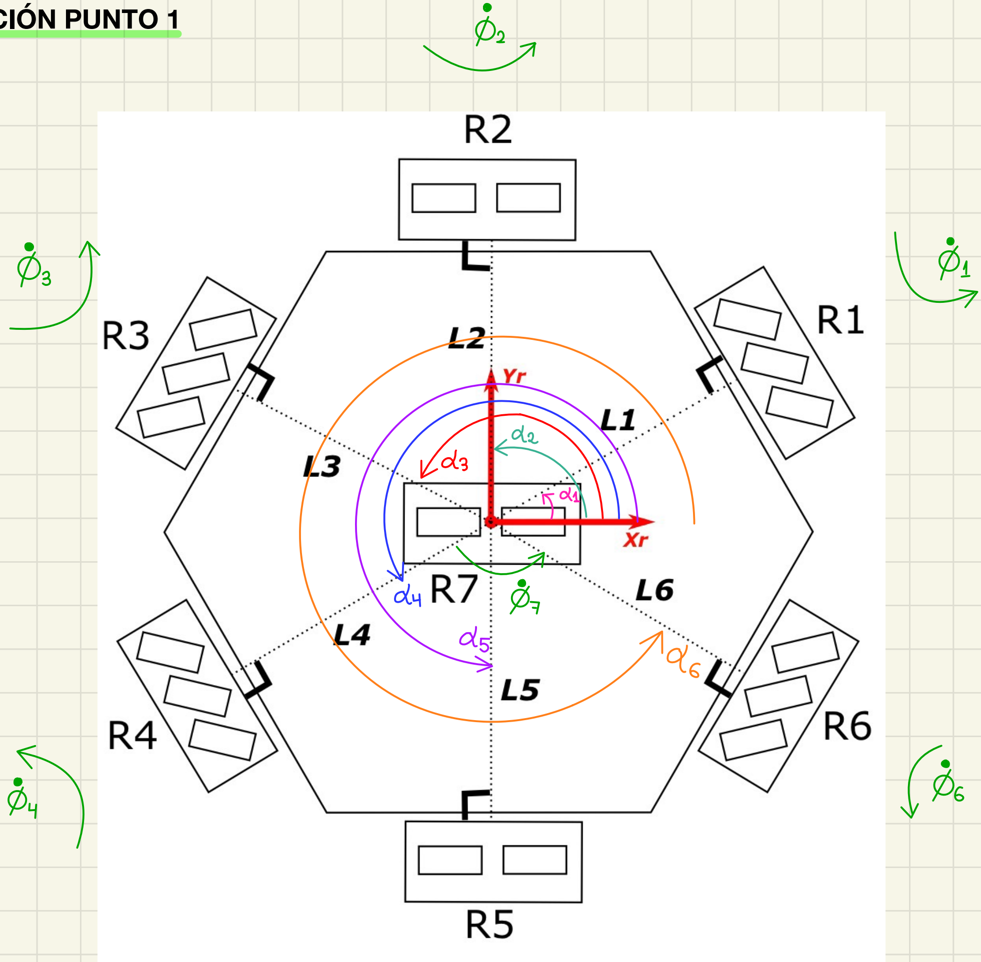


Figura 1: Robot móvil a diseñar.

$\dot{\phi}_5$

	α_i	β_i	r_i	γ_i	l_i
R_{1sw}	$\frac{\pi}{6}$	0	0,03m	$\frac{7\pi}{36}$	0,25m
R_{2sw}	$\frac{\pi}{2}$	0	0,02m	0	0,25m
R_{3sw}	$\frac{5\pi}{6}$	0	0,03m	$\frac{7\pi}{36}$	0,25m
R_{4sw}	$\frac{7\pi}{6}$	0	0,03m	$\frac{7\pi}{36}$	0,25m
R_{5sw}	$\frac{3\pi}{2}$	0	0,02m	0	0,25m
R_{6sw}	$\frac{11\pi}{6}$	0	0,03m	$\frac{7\pi}{36}$	0,25m
R_{7sw}	0	$\frac{\pi}{2}$	0,02m	0	0m

Nota: existen varias configuraciones válidas para la rueda 7 (yo elegí "$\alpha_7 = 0$" y "$\beta_7 = \frac{\pi}{2}$")

¡Generamos tabla de parámetros!

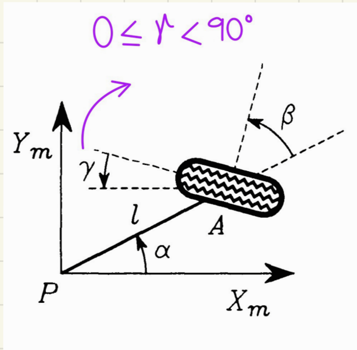


Figure 7.4: Swedish wheel.

* Ahora encontramos restricciones generales para el modelo...

$$\boxed{\begin{aligned} J_1(\beta_s,\beta_c)R^T(\vartheta)\dot{\xi} + J_2\dot{\varphi} &= 0 \\ C_1(\beta_s,\beta_c)R^T(\vartheta)\dot{\xi} + C_2\dot{\beta}_c &= 0. \end{aligned}}$$

$\hookrightarrow \dot{z}$

$$\left(-\sin(\alpha+\beta+\gamma) \quad \cos(\alpha+\beta+\gamma) \quad l\cos(\beta+\gamma)\right) R^T(\vartheta)\dot{\xi} + r\cos\gamma\dot{\varphi} = 0 \qquad (7.9)$$

$\hookrightarrow \dot{z}$

$$\boxed{C_1 = C_2 = 0}$$

$\hookrightarrow$ No aplican en este modelo

$$J_2 = \begin{bmatrix} r_1\cdot c(\gamma_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2\cdot c(\gamma_2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_3\cdot c(\gamma_3) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_4\cdot c(\gamma_4) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_5\cdot c(\gamma_6) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_6\cdot c(\gamma_7) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_7\, c(\gamma_7) \end{bmatrix}$$

$$J_1 = \begin{bmatrix} -s(\alpha_1+\beta_1+\gamma_1) & c(\alpha_1+\beta_1+\gamma_1) & l_1\cdot c(\beta_1+\gamma_1) \\ -s(\alpha_2+\beta_2+\gamma_2) & c(\alpha_2+\beta_2+\gamma_2) & l_2\, c(\beta_2+\gamma_2) \\ -s(\alpha_3+\beta_3+\gamma_3) & c(\alpha_3+\beta_3+\gamma_3) & l_3\cdot c(\beta_3+\gamma_3) \\ -s(\alpha_4+\beta_4+\gamma_4) & c(\alpha_4+\beta_4+\gamma_4) & l_4\cdot c(\beta_4+\gamma_4) \\ -s(\alpha_5+\beta_5+\gamma_5) & c(\alpha_5+\beta_5+\gamma_5) & l_5\cdot c(\beta_5+\gamma_5) \\ -s(\alpha_6+\beta_6+\gamma_6) & c(\alpha_6+\beta_6+\gamma_6) & l_6\cdot c(\beta_6+\gamma_6) \\ -s(\alpha_7+\beta_7+\gamma_7) & c(\alpha_7+\beta_7+\gamma_7) & l_7\cdot c(\beta_7+\gamma_7) \end{bmatrix}$$

* Como NO hay nada que restringe mis grados de libertad...

$$\boxed{\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

$$E = -J_2^{-1}\cdot J_1$$

$\hookrightarrow$ Matlab

$$\boxed{\dot{z} = R(\theta)\Sigma\cdot\varepsilon} \qquad \boxed{\dot{\phi} = E\cdot\Sigma\cdot\varepsilon}$$

$\hookrightarrow$ Matlab $\qquad$ $\hookrightarrow$ Matlab

* Se procede a realizar procedimiento en Software MATLAB...

```
% POINT 1: OMNIDIRECTIONAL 7 WHEEL ROBOT (CANUDAS APPROACH)
% SANTIAGO GARCIA ARANGO // ROBOTICS FINAL
clc; clear variables; format short;
addpath(genpath("../"));  % Acces upper folder functions
% sympref('FloatingPointOutput',true);

% Add necessary symbolic variables for the solution
syms theta x_d y_d theta_d Vx Vy W

% Define each sub-parameter of the wheels (alfa, beta, radius, gamma,
 L)
```

Definimos Tabla

```
a1 = pi/6;     b1 = 0;     r1 = 0.03;  gamma1 = 7*pi/36;  l1 = 0.25;
a2 = pi/2;     b2 = 0;     r2 = 0.02;  gamma2 = 0;        l2 = 0.25;
a3 = 5*pi/6;   b3 = 0;     r3 = 0.03;  gamma3 = 7*pi/36;  l3 = 0.25;
a4 = 7*pi/6;   b4 = 0;     r4 = 0.03;  gamma4 = 7*pi/36;  l4 = 0.25;
a5 = 3*pi/2;   b5 = 0;     r5 = 0.02;  gamma5 = 0;        l5 = 0.25;
a6 = 11*pi/6;  b6 = 0;     r6 = 0.03;  gamma6 = 7*pi/36;  l6 = 0.25;
a7 = 0;        b7 = pi/2;  r7 = 0.02;  gamma7 = 0;        l7 = 0;

% Constraints for the second part of equation
J2 = [r1*cos(gamma1), 0, 0, 0, 0, 0, 0;
      0, r2*cos(gamma2), 0, 0, 0, 0, 0;
      0, 0, r3*cos(gamma3), 0, 0, 0, 0;
      0, 0, 0, r4*cos(gamma4), 0, 0, 0;
      0, 0, 0, 0, r5*cos(gamma5), 0, 0;
      0, 0, 0, 0, 0, r6*cos(gamma6), 0;
      0, 0, 0, 0, 0, 0, r7*cos(gamma7)]

% Constraints for the first part of the equation
J1 = [-sin(a1+b1+gamma1), cos(a1+b1+gamma1), l1*cos(b1+gamma1);
      -sin(a2+b2+gamma2), cos(a2+b2+gamma2), l2*cos(b2+gamma2);
      -sin(a3+b3+gamma3), cos(a3+b3+gamma3), l3*cos(b3+gamma3);
      -sin(a4+b4+gamma4), cos(a4+b4+gamma4), l4*cos(b4+gamma4);
      -sin(a5+b5+gamma5), cos(a5+b5+gamma5), l5*cos(b5+gamma5);
      -sin(a6+b6+gamma6), cos(a6+b6+gamma6), l6*cos(b6+gamma6);
      -sin(a7+b7+gamma7), cos(a7+b7+gamma7), l7*cos(b7+gamma7)]

% MANUAL FIXES (SMALL NUMBERS)--- This is because they could affect
 null space-base
J1(2,2) = 0;
J1(5,2) = 0;
J1(7,2) = 0

SIGMA = eye(3)  % Sigma couple matrix doesn't have any restrictions on
 this model
E = -J2^(-1)*J1
TM = transformation([0, 0, 1], [0, 0, 0], theta, 1);
R_theta = TM(1:3, 1:3);


Z_dot = R_theta*SIGMA*[Vx; Vy; W]
phi_dot = E*SIGMA*[Vx; Vy; W]
```

```
J2 =

    0.0246         0         0         0         0         0         0
         0    0.0200         0         0         0         0         0
         0         0    0.0246         0         0         0         0
         0         0         0    0.0246         0         0         0
         0         0         0         0    0.0200         0         0
         0         0         0         0         0    0.0246         0
         0         0         0         0         0         0    0.0200


J1 =

   -0.9063    0.4226    0.2048
   -1.0000    0.0000    0.2500
    0.0872   -0.9962    0.2048
    0.9063   -0.4226    0.2048
    1.0000   -0.0000    0.2500
   -0.0872    0.9962    0.2048
   -1.0000    0.0000         0
```

Aplico restricción de que números pequeños ($\# < 10 * 10^{-10}$), sea cero. (Evitar errores de aproximación).

```
J1 =

   -0.9063    0.4226    0.2048
   -1.0000         0    0.2500
    0.0872   -0.9962    0.2048
    0.9063   -0.4226    0.2048
    1.0000         0    0.2500
   -0.0872    0.9962    0.2048
   -1.0000         0         0


SIGMA =

     1     0     0
     0     1     0
     0     0     1


E =

   36.8799  -17.1974   -8.3333
   50.0000         0  -12.5000
   -3.5466   40.5376   -8.3333
  -36.8799   17.1974   -8.3333
  -50.0000         0  -12.5000
    3.5466  -40.5376   -8.3333
   50.0000         0         0


Z_dot =
```

$$\dot{z} = \begin{bmatrix} Vx*cos(theta) - Vy*sin(theta) \\ Vy*cos(theta) + Vx*sin(theta) \\ W \end{bmatrix}$$

Resultados finales del modelo de postura y de velocidad en ruedas

```
phi_dot =
```

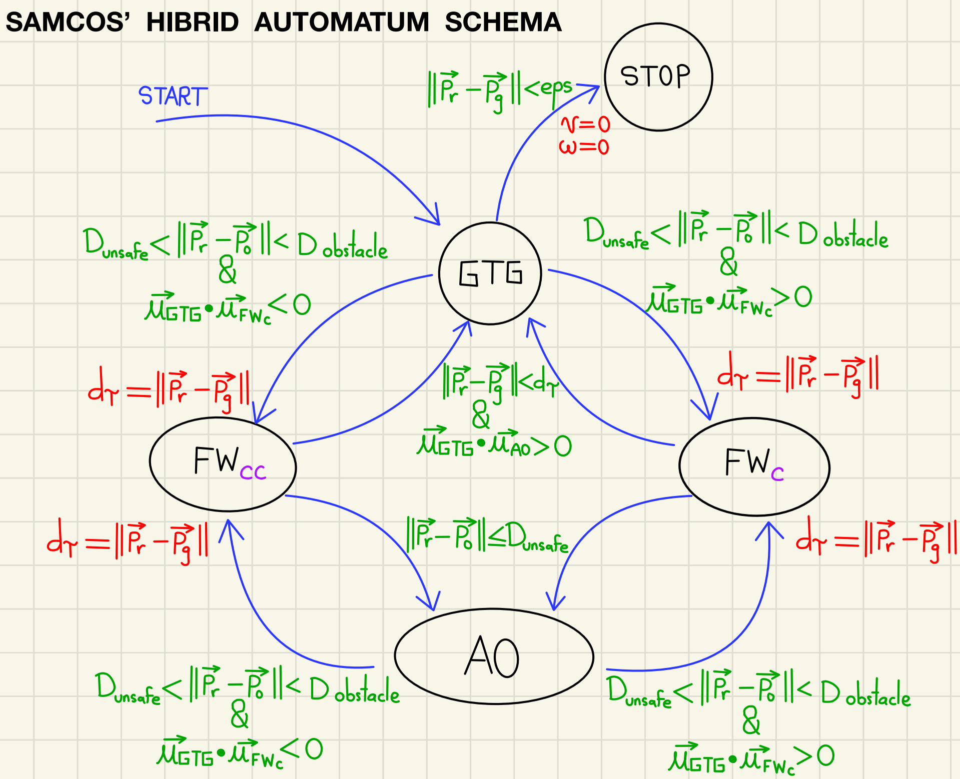
$$\dot{\phi} = \begin{bmatrix} 36.8799*Vx - 17.1974*Vy - 8.3333*W \\ 50*Vx - 12.5000*W \\ 40.5376*Vy - 3.5466*Vx - 8.3333*W \\ 17.1974*Vy - 36.8799*Vx - 8.3333*W \\ - 50*Vx - 12.5000*W \\ 3.5466*Vx - 40.5376*Vy - 8.3333*W \\ 50*Vx \end{bmatrix}$$

Published with MATLAB® R2020a

SOLUCIÓN PUNTO 2

El primer paso para realizar un sistema completo de navegación, teniendo en cuenta una estructura de autómata híbrido, es plantear cómo va a funcionar el sistema en términos generales... es decir, cómo será su funcionamiento desde un punto de vista general de poder ejecutar sus acciones con base en comportamientos deseados.

Para esto, se planteará una estructura de un autómata híbrido con las siguientes condiciones (que pueda realizar 3 tareas principales: GoToGoal, FollowWall y AvoidObstacles).

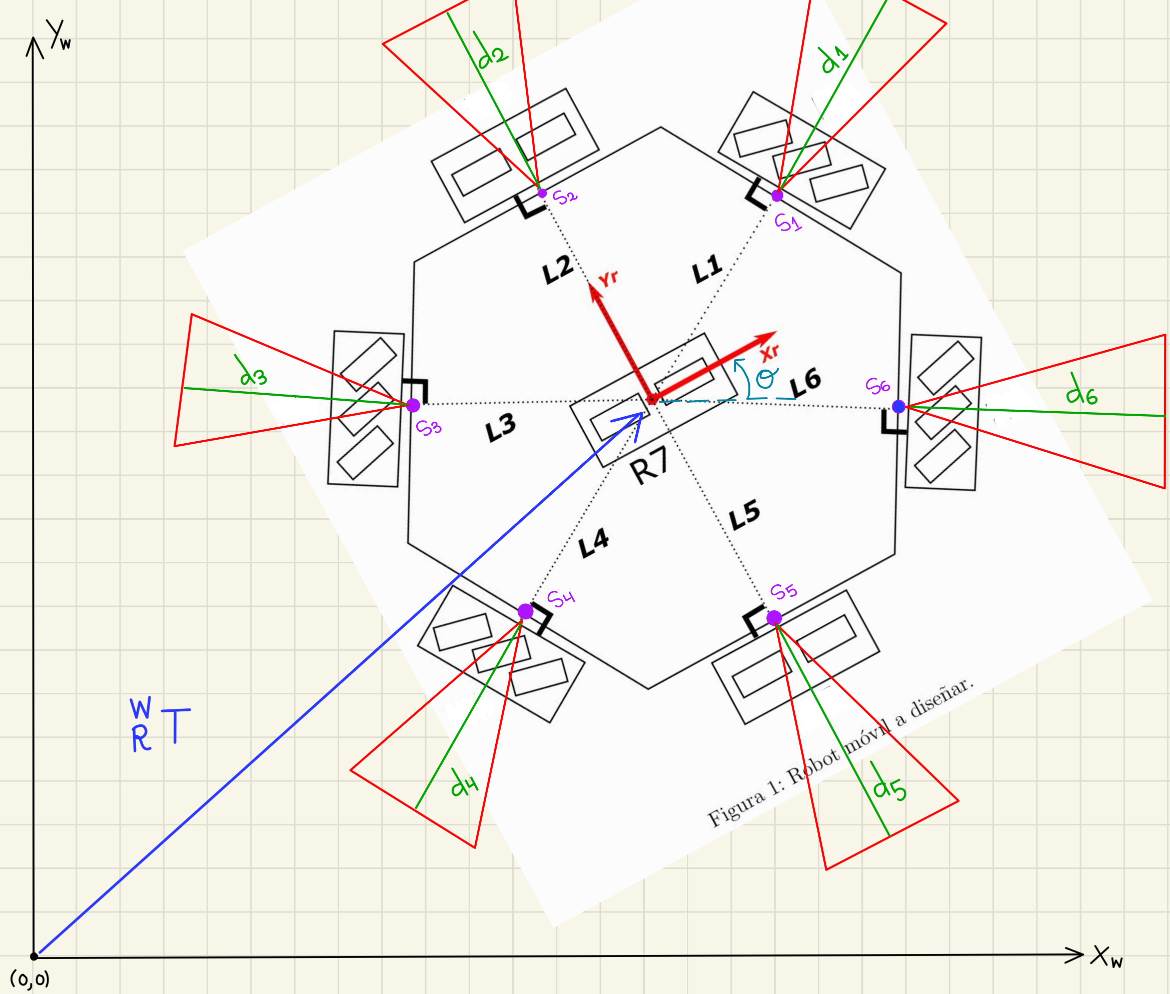


El punto de partida para realizar este esquema, es tener una noción de los obstáculos, pues para poder seguir paredes o evadir obstáculos, se debe tener in sistema de medición externo, que permita detectar aproximadamente estos.

A continuación, se procede a explicar cómo sería su implementación.

NOTA: se irá explicando cada uno de estos modos y al final se explicará cómo los junté.

SENSORES A IMPLEMENTAR



Habían dos opciones de colocar los sensores primordiales: en las aristas o en los vértices. Yo elegí la de las aristas, porque puede tener una instalación física más sencilla, mientras que garantizar que estos se instalen correctamente en una esquina del robot y que queden alineados de forma "radial", es más complejo desde el punto de vista práctico.

Cada uno de estos sensores, dará directamente una distancia "di" a su punto de medición captado, implicando que aunque tenemos información de estos, se debe llevar a un sistema de referencia conocido!

En la siguiente página explico esto.

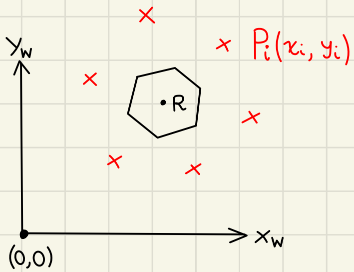
En primer lugar, se debe transformar estas distancias de los sensores infrarrojos implementados, al sistema de referencia del robot (en este caso R). Esto para poder tener los puntos (Xi,Yi) de todos 6 los sensores desde el Robot Frame.

$$\boxed{{}^{R}d_i = {}^{R}_{si}T \; {}^{si}d_i}$$

Teniendo planteada la forma de obtener cada una de las distancias de los sensores desde el sistema de referencia R del CM del robot, se debe ahora encontrar la equivalencia de estas desde el sistema de referencia global (mundo).

$$\boxed{{}^{W}d_i = {}^{W}_{R}T \cdot {}^{R}d_i}$$

Traslación (x,y) y Rotación (θ)



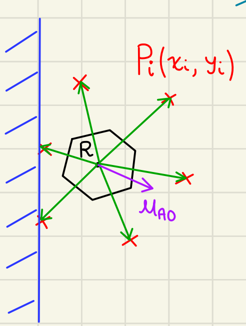
En esta instancia, se deberían tener cada uno de los puntos mostrados en el marco de referencia global, los cuales serán de gran utilidad para nuestro procedimiento a realizar, pues ahora tenemos una noción del mundo con un sistema determinado.

AVOID OBSTACLES

Avoid Obstacles, junto con GoToGoal, hacen parte de las herramientas más importantes de la robótica móvil, pues son “el dúo dinámico” para poder realizar la navegación “autónoma”.

En este robot, utilizaré la forma más sencilla y práctica de evadir obstáculos, la cual es a través de una sumatoria ponderada de los vectores obtenidos por cada sensor.

Al ser simétricos, NO hace falta unos pesos ponderados!



$$\boxed{u_{AO} = \sum \vec{d_i}}$$

Nota: es importante aclarar, que según nuestro modelo de autómata híbrido, únicamente se va a llegar a AvoidObstacles, cuando alguna de las distancias “d_i” sea menor a una distancia segura, es decir, cuando estemos MUY cerca aun objeto.

Al calcular este vector, siempre estará con un vector resultante opuesto a la pared, por la configuración de dichos sensores.

Cuando se tenga el vector de Avoid Obstacles , se debe proceder a generar una acción de control para alejarse de la pared, a través de este...

Un ejemplo de esta acción puede ser:

Nota: realmente se pueden generar modelos que apliquen una rotación angular con "ω", pero lo dejaré simple!

$$u_{AO} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \longrightarrow \mathcal{E} = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 \ \frac{m}{s} \\ \sqrt{2}/2 \ \frac{m}{s} \\ 0 \ \frac{rad}{s} \end{bmatrix}$$

Dependiendo de rangos de Voltajes / Corrientes, se debe escalar este vector por un "α".

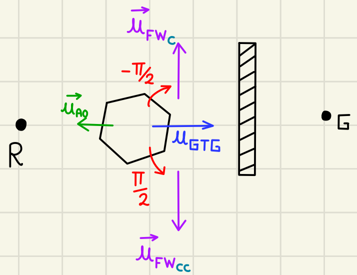
Y gracias al procedimiento de Canudas previo, podemos aplicar estas condiciones con los modelos de postura y velocidad en las ruedas respectivo, para mandar las acciones de control aproximadas a cada una de las ruedas:

$$\dot{\phi} = \begin{bmatrix} 36.8799*Vx - 17.1974*Vy - 8.3333*W \\ 50*Vx - 12.5000*W \\ 40.5376*Vy - 3.5466*Vx - 8.3333*W \\ 17.1974*Vy - 36.8799*Vx - 8.3333*W \\ -50*Vx - 12.5000*W \\ 3.5466*Vx - 40.5376*Vy - 8.3333*W \\ 50*Vx \end{bmatrix}$$

FOLLOW WALL

Para proceder a seguir la pared (FollowWall), me aprovecharé que ya tenemos el vector de Avoid Obstacles, y por lógica, se puede visualizar que la pared está aproximadamente con un desfase ortogonal del mismo, es decir, se puede puede girar este vector AO un ángulo de +- pi/2 para generar un vector estimado de seguimiento de dicha pared.

Esto implica que podremos seguir la pared en las dos direcciones mostradas.



$$\vec{u}_{FW_c} = \vec{u}_{AO} \cdot R(-\pi/2)$$

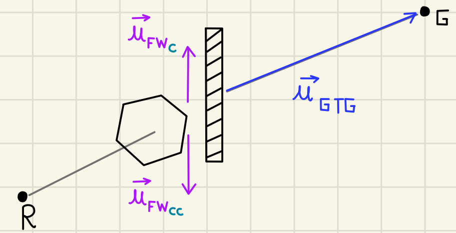
$$\vec{u}_{FW_{cc}} = \vec{u}_{AO} \cdot R(+\pi/2)$$

OJO: tener presente dirección de nuestro $\vec{u}_{AO}$.

Teniendo en cuenta dicho planteamiento, se debe tener un método de selección de qué lado elegir para el movimiento neto del robot.

La forma más práctica de realizar esta elección, es a través de un producto punto entre el vector de follow wall (FW) en un sentido y el vector de go to goal (GTG).

Según el resultado (signo), se podrá determinar cuál de los dos sentidos tiende a estar direccionado hacia el Goal, y por ende, estimar que se debe seguir la pared por el lado derecho o izquierdo...
Nota: esto NO garantiza nada, pero es una alternativa utilizada en robots diferenciales (con lo poquito que conocemos en cualquier instante del movimiento del robot).



¿Cuál dirección elegir?

* Elegimos dirección a seguir:

$$\vec{u}_{GTG} \cdot \vec{u}_{FW_c} > 0 \longrightarrow \vec{u}_{FW_c}$$

$$\vec{u}_{GTG} \cdot \vec{u}_{FW_c} < 0 \longrightarrow \vec{u}_{FW_{cc}}$$

Es por esto, que el vector resultante se podrá obtener com base en el vector de evadir obstáculos mostrado anteriormente... y una transformación simple, luego de determinar hacia qué lado seguir la pared.

Un ejemplo cualquiera:

Nota: realmente se pueden generar modelos que apliquen una rotación angular con "ω", pero lo dejaré simple!

$$u_{AO} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \xrightarrow[\text{Rotación } \pi/2]{} \varepsilon = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{2}}{2}\,\frac{m}{s} \\ \frac{\sqrt{2}}{2}\,\frac{m}{s} \\ 0\,\frac{rad}{s} \end{bmatrix}$$

Dependiendo de rangos de Voltajes/Corrientes, se debe escalar este vector por un "α"

Y gracias al procedimiento de Canudas previo, podemos aplicar estas condiciones con los modelos de postura y velocidad en las ruedas respectivo, para mandar las acciones de control aproximadas a cada una de las ruedas:

$$\dot{\phi} = \begin{bmatrix} \frac{36.8799*Vx - 17.1974*Vy - 8.3333*W}{50*Vx - 12.5000*W} \\ \frac{40.5376*Vy - 3.5466*Vx - 8.3333*W}{17.1974*Vy - 36.8799*Vx - 8.3333*W} \\ \frac{- 50*Vx - 12.5000*W}{3.5466*Vx - 40.5376*Vy - 8.3333*W} \\ 50*Vx \end{bmatrix}$$

ALGUNOS DETALLES IMPORTANTES FINALES

Aunque ya tengo un modelo de seguir paredes y evadir obstáculos (FW, AO), realmente estos deben funcionar con las condiciones del autómata híbrido mostrado unas páginas atrás...

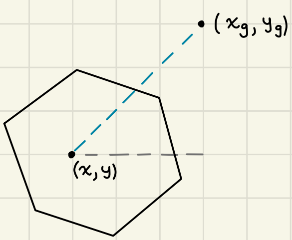
Para poder lograr esto, se debe generar los switcheos discretos entre cada modo de operación.

1. El primer detalle a tener en cuenta, es que LA ÚNICA manera de entrar en el modo de evadir obstáculos, es a través de estar muy cerca a uno de los obstáculos... es decir, siempre se entrará primero al modo follow wall y luego, dependiendo del resultado de este, se puede o no entrar a AviidObstacles (solamente en emergencias o al estar a una distancia “Unsafe”). La salida de AvoidObstacles, podrá ser realizada a través de volver a estar a una distancia “Safe” del objeto, donde se debe volver a follow wall y aplicar un Reset al “TAU” (lo explicaré abajo).

2. El segundo detalle importante, es que se debe saber cuándo se debe dejar de seguir las paredes... esto se logrará ÚNICAMENTE, cuando se haya hecho progreso neto... esto significa, que el robot se encuentre más cerca del goal, que antes de comenzar a seguir paredes (pues de lo contrario, se quedaría en un loop infinito siguiendo la pared). Esto lo demarqué en el esqumea del autómata híbrido como un “TAU” a través de un reseteo.

3. El tercer detalle importante, es que el autómata debe poder parar de forma efectiva... para poder lograr esto, cuando esté en Follow Wall, se deberá poder determinar el error absoluto entre el punto actual estimado y el goal. Si este error es menor a un epsilon, se asume que se llegó (porque de lo contrario, se puede quedar generando acciones de control indeseadas, o incluso un switcheo del fenómeno de Zeno tipo 2 por no poder nunca llegar al cero absoluto en el error.

4. Por último, el GoToGoal, puede ser aplicado similar al que vimos en clase, o incluso un nuevo esquema de control agregando un grado de libertad adicional al sistema (w).
Lo dejaré como una variante del visto en clase.



$$SP = \vec{r} = [x_g, y_g]$$

$$\phi_d = \text{atan2}(y_g - y \,;\, x_g - x)$$ → Permite acotar ϕs entre $-\pi$ y π !

$$e = \phi_d - \phi$$

Ejemplo...

$$\mathcal{E} = \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} x_g - x_R \\ y_g - y_R \\ \omega \end{bmatrix}$$

(Sólo si necesario)