

MODELO CINEMÁTICO ROBOTS MÓVILES

Robótica Aplicada

Mtro. Huber Girón Nieto

Otoño 2023



Departamento
de Ciencias
e Ingenierías /

CONTENIDO

2

1. Robots móviles con ruedas.
2. Plataforma de experimentación.
3. Visión por computadora.
4. Estrategia de control: Partícula.
5. Modelo Cinemático.
6. Extensión dinámica.
7. Estrategia de control: Robot uniciclo.
8. Proyecto: Robot móvil.

1. ROBOTS MÓVILES CON RUEDAS

ROBOTS MÓVILES

- Los robots móviles, son robots que pueden moverse de un lugar a otro autónomamente, sin la necesidad de asistencia de operadores humanos externos.
- A diferencia de la mayoría de los robots industriales que solo pueden moverse en un específico espacio de trabajo, los robots móviles tienen la capacidad de moverse libremente sin un espacio de trabajo predefinido para cumplir con sus tareas.



Figura 1.1. Robots móviles con ruedas.



Figura 1.2. Robots móviles con piernas.



Figura 1.3. Robots móviles Aéreos.



Figura 1.4. Robots móviles Acuáticos.

ROBOTS MÓVILES CON RUEDAS

- La maniobrabilidad de los robots con ruedas (WMR – Wheeled mobile robot) depende del tipo de ruedas y la configuración de estas.



Figura 1.5. Tipos de ruedas. a) Fija, b) Castor, c) Sueca, d) Esférica

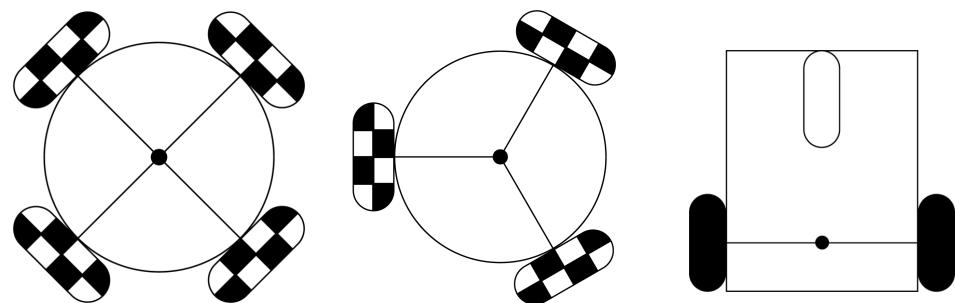


Figura 1.6. Diferentes configuraciones de robots móviles.

TIPOS DE RUEDA

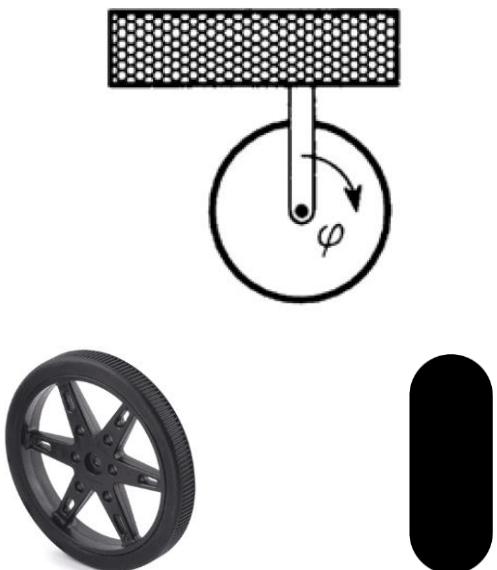


Figura 1.7. Rueda Fija (1GDL)

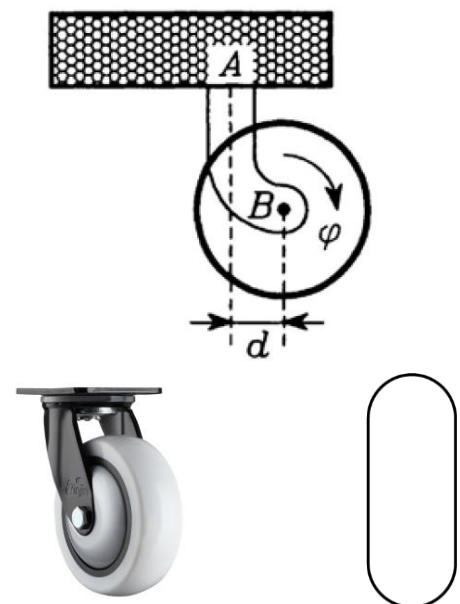


Figura 1.8. Rueda Loca o tipo Castor (2GDL).

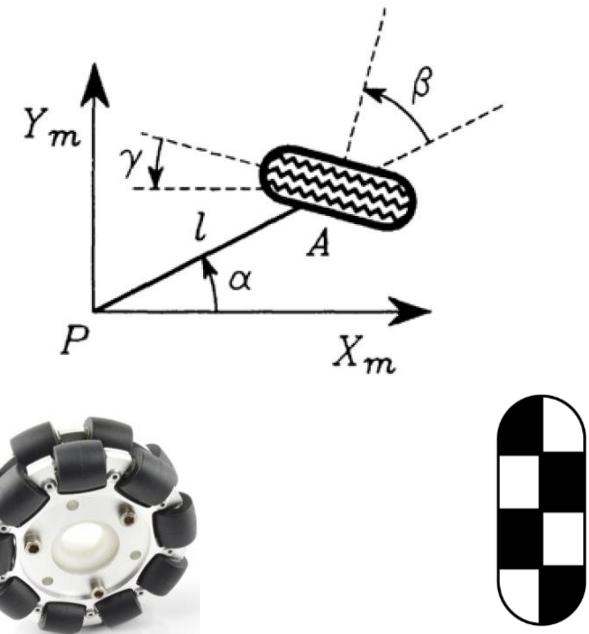


Figura 1.9. Rueda Omnidireccional o Sueca (2GDL).

CONFIGURACIONES

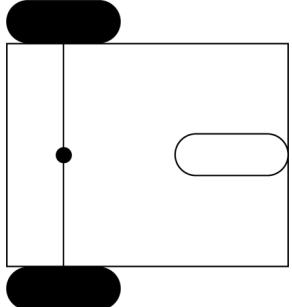


Figura 1.10. WMR 3
llantas: 2 fijas y 1 castor.

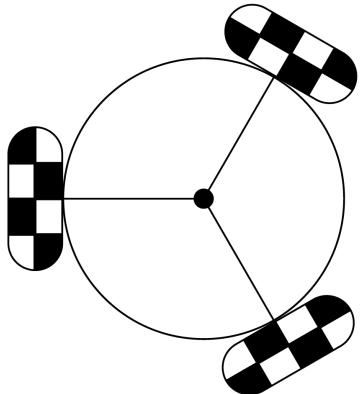


Figura 1.11. WMR 3 llantas:
3 omnidireccionales

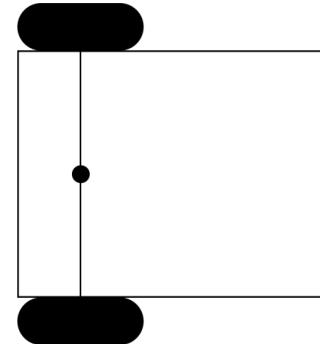


Figura 1.12. WMR 2
llantas: 2 fijas.

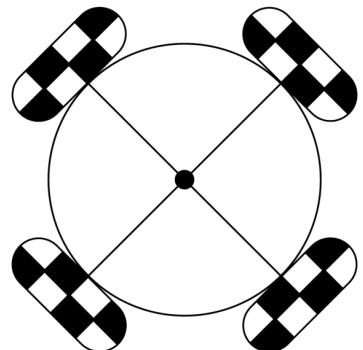


Figura 1.13. WMR 4 llantas: 4
omnidireccionales

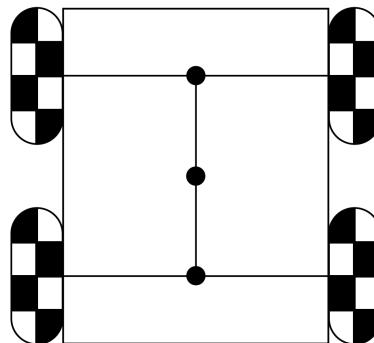
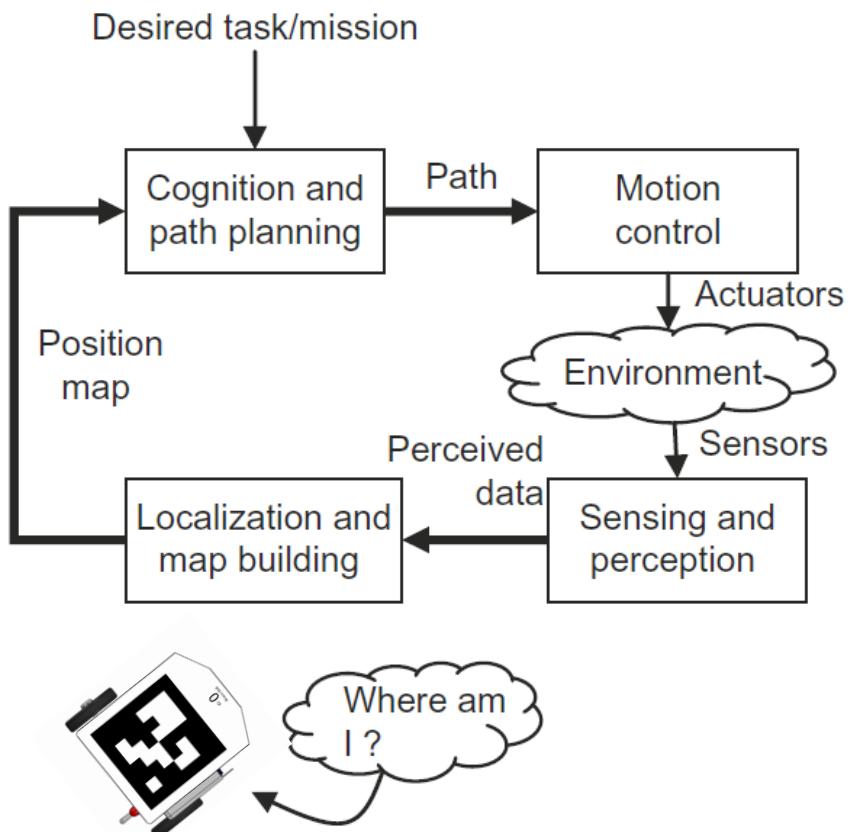


Figura 1.14. WMR 4 llantas: 4
omnidireccionales

2. PLATAFORMA DE EXPERIMENTACIÓN

LOCALIZACIÓN Y POSICIÓN DE ROBOTS MÓVILES



- Los robots móviles deben tener la capacidad de percibir su entorno mediante el uso de sensores para determinar en un principio su localización y posteriormente determinar el destino, las tareas y el camino necesario para obtener la meta.

PLATAFORMA DE EXPERIMENTACIÓN

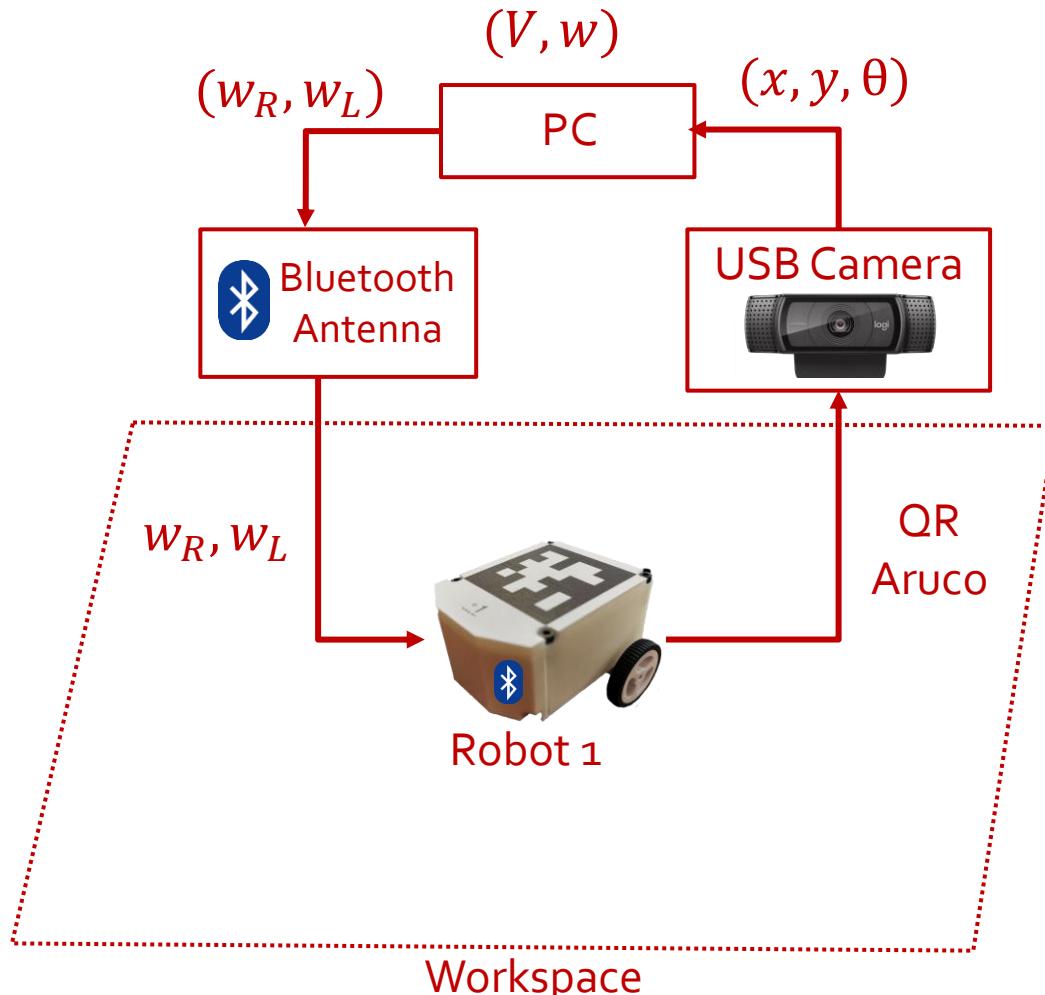


Figura 2.2. Plataforma de experimentación.

- Para este curso se utilizará un entorno de trabajo con una computadora personal (PC) que tenga antena Bluetooth y una cámara USB.
- Los robots móviles igualmente contarán con un módulo Bluetooth (HC-05) y una batería para brindarles autonomía.
- Las estrategias de control y comunicación en la PC se programarán mediante Python en el entorno de Visual Studio Code.
- Las estrategias de control y comunicación en el robot móvil se programarán en una tarjeta de desarrollo utilizando el entorno de Arduino.

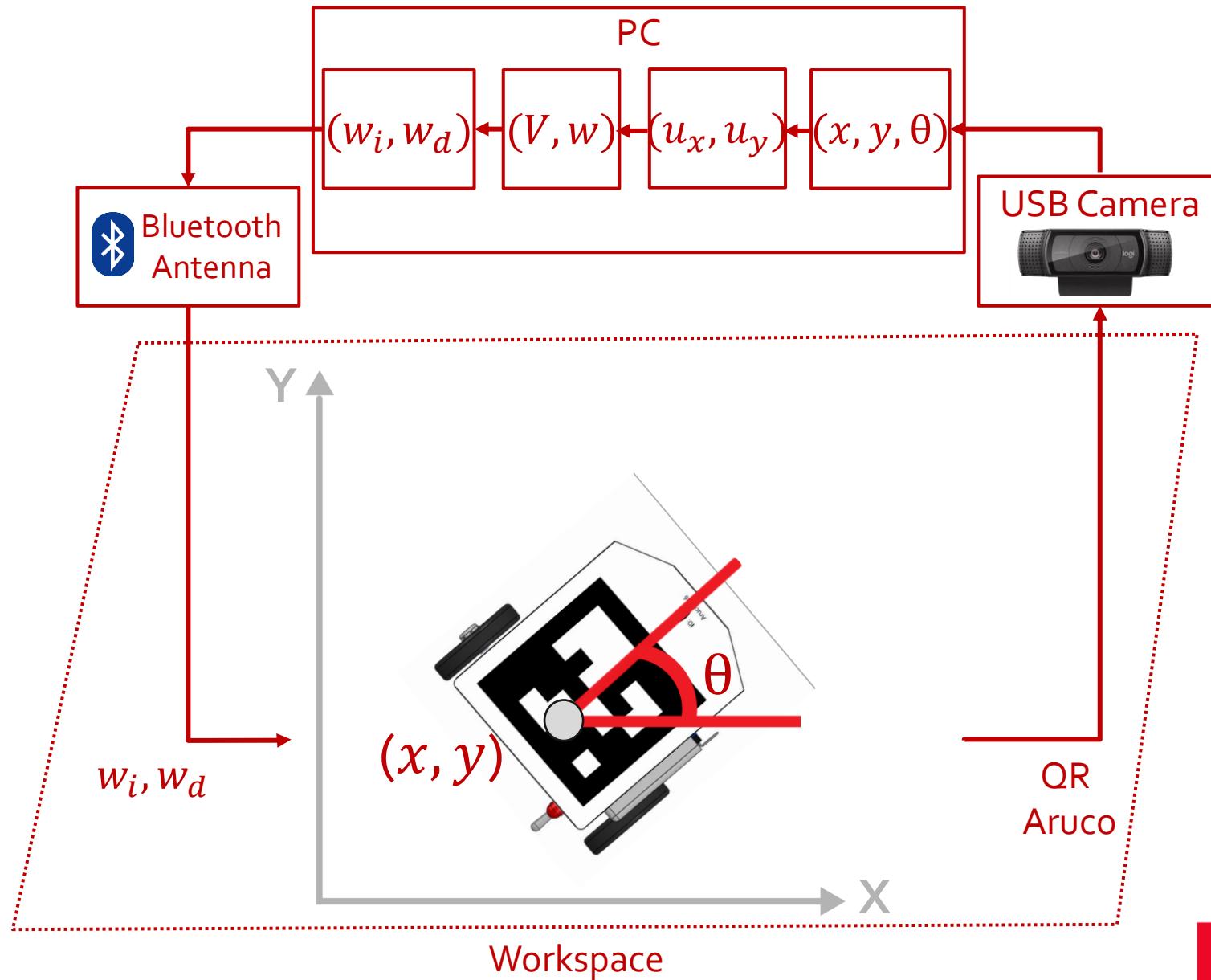


Figura 2.3. Plataforma de experimentación.

EJEMPLO: ROBOT UNICICLO

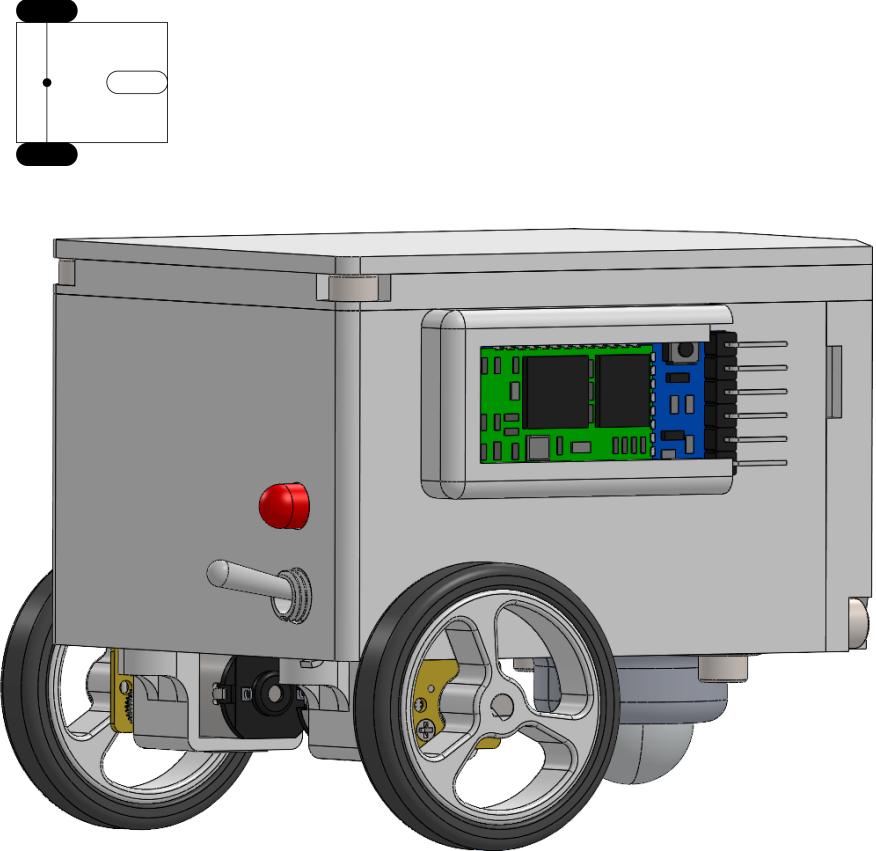


Figura 2.4. Robot tipo uniciclo de 3 llantas: 2 fijas y 1 castor.

12

Componente	#	Foto
Pololu Micro motorreductor N20 600rpm	2	
Llanta Pololu N20 34X7mm	2	
Arduino Pro Micro 5v 16Mhz	1	
Batería LIPO 7.4V 900mAh	1	

Tabla 2.1. Lista de componentes de robot uniciclo.

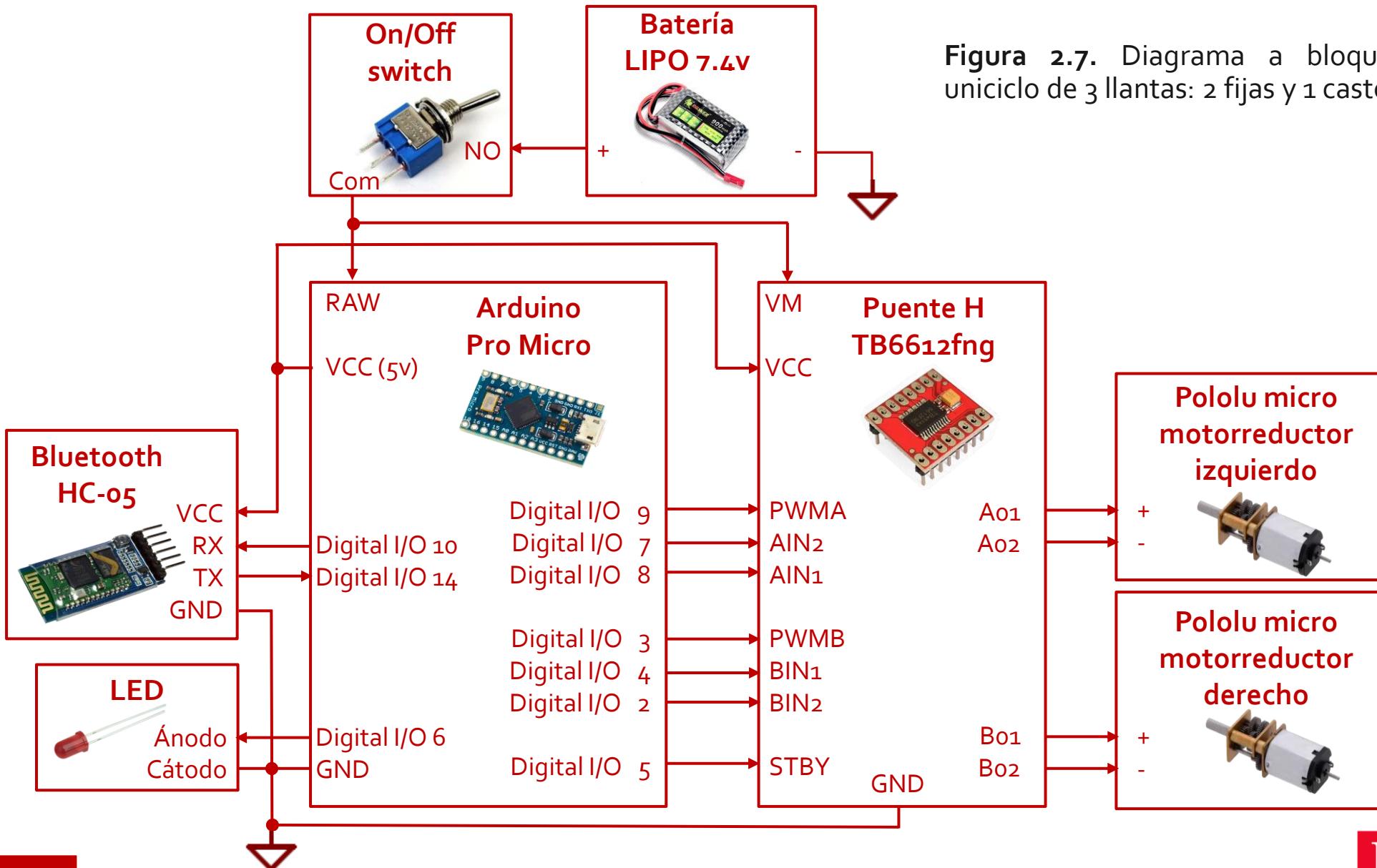


Figura 2.7. Diagrama a bloques de robot uniciclo de 3 llantas: 2 fijas y 1 castor.

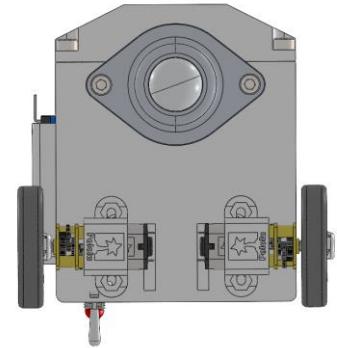


Figura 2.5. Vista inferior robot uniciclo.

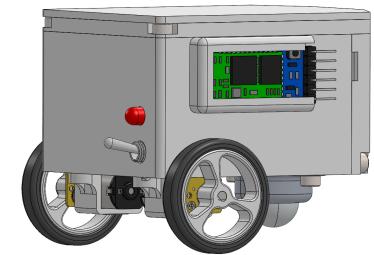


Figura 2.6. Vista isométrica robot uniciclo.

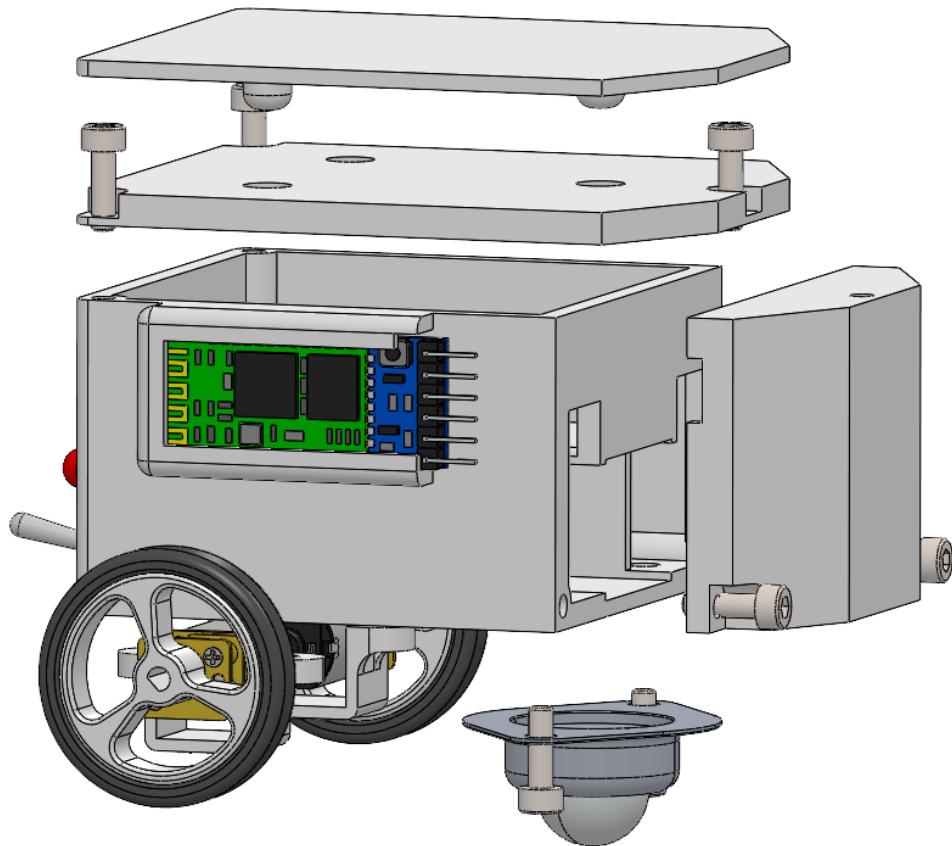


Figura 2.8. Ensamble en explosión robot uniciclo.

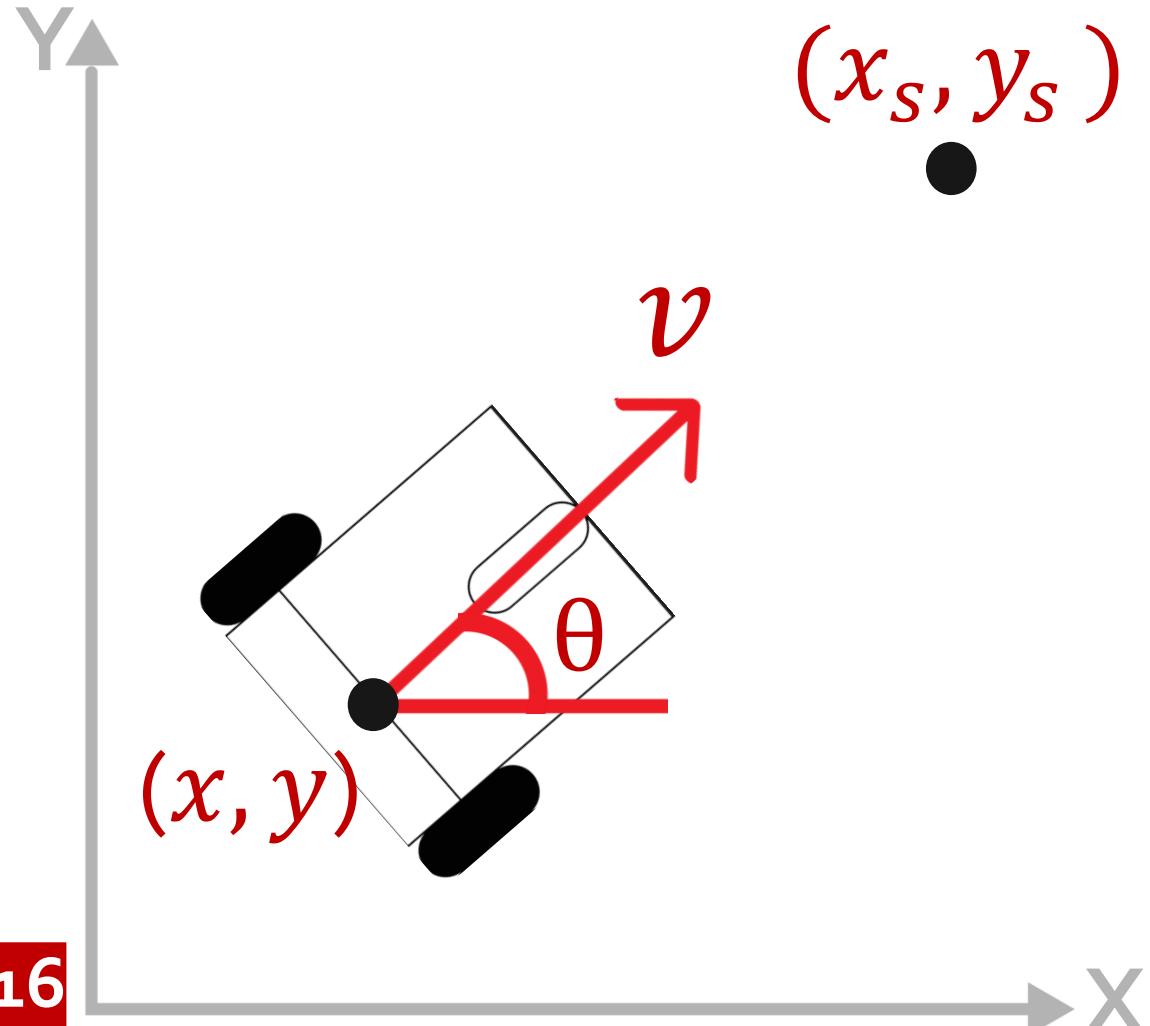
Repository de Github:

<https://github.com/HuberGiron/Robot-uniciclo-Ibero-2>

- 1. Ensamble en Solidworks.
- 2. Archivos STL para impresión 3D.
- 3. Código Arduino.
- 4. Estrategias de control en Python.

3. VISION POR COMPUTADORA

LOCALIZACIÓN Y POSICIÓN DE ROBOTS MÓVILES



$(x, y) \rightarrow$ posición actual
 $\theta \rightarrow$ orientación

$(x_s, y_s) \rightarrow$ posición deseada
(star point)

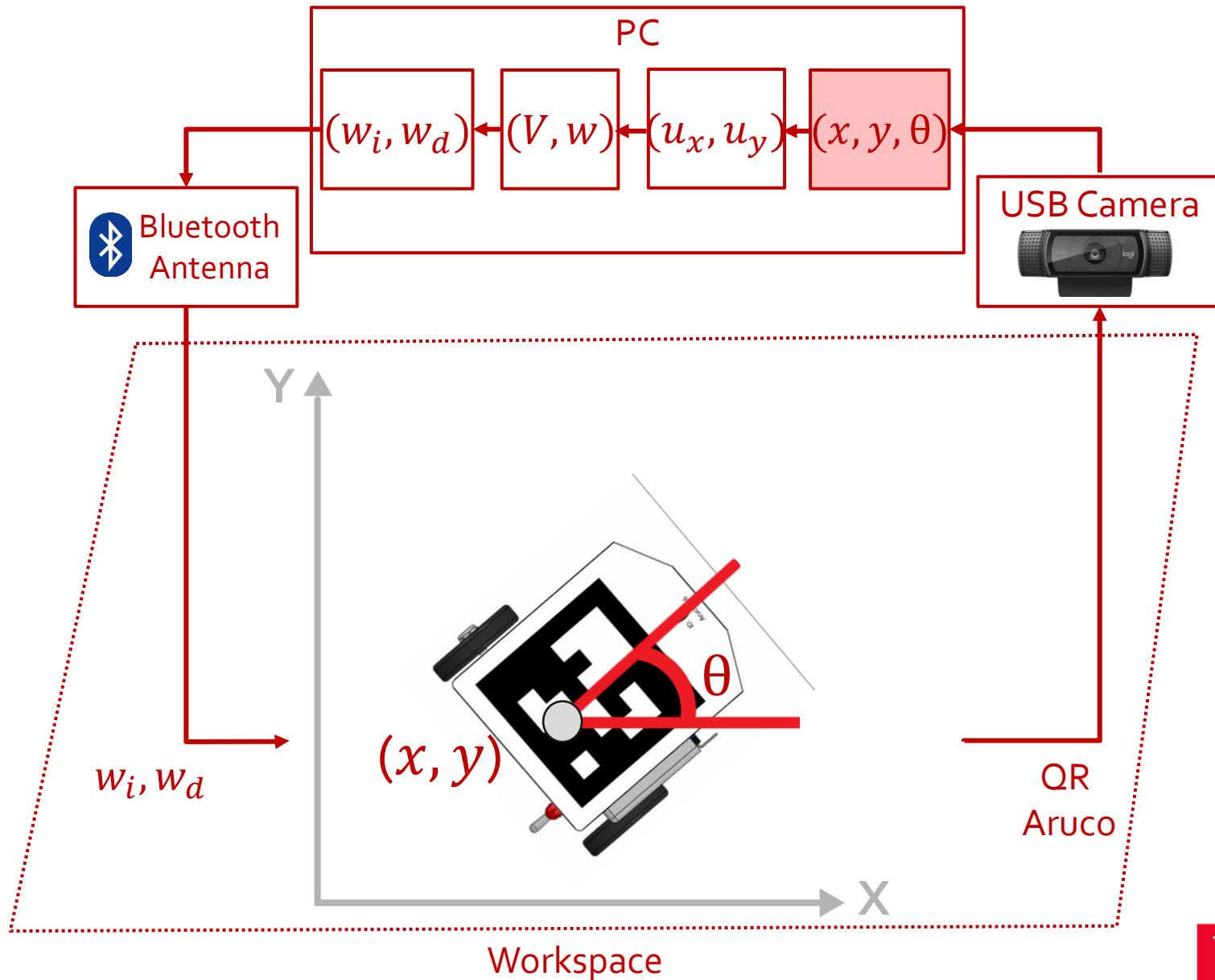


Figura 3.o. Plataforma de experimentación.

VISIÓN POR COMPUTADORA

- La visión por computadora estudio los métodos para reconocer y localizar objetos del ambiente mediante el procesamiento de imágenes.
- Al momento de determinar el hardware, es importante considerar la resolución del equipo, la luz y el enfoque necesario del ambiente para captar la imagen.



Figura 3.1. Webcam USB.



Figura 3.2. VICON MOCAP System.

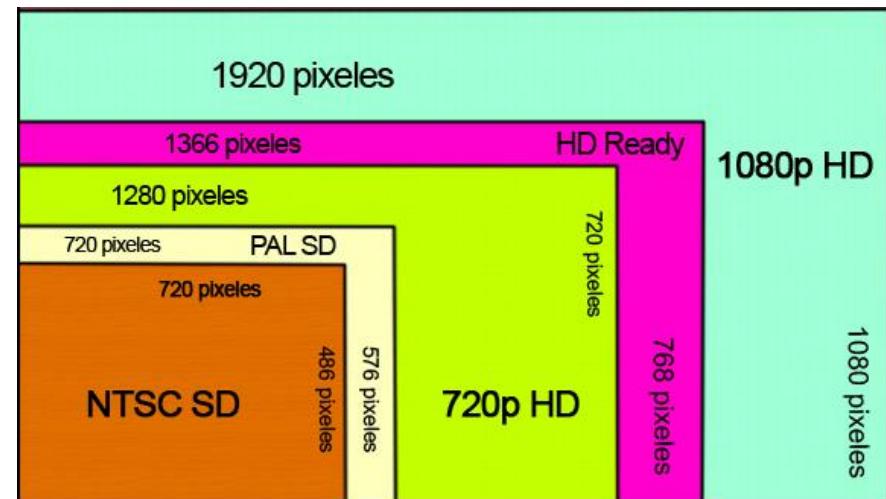


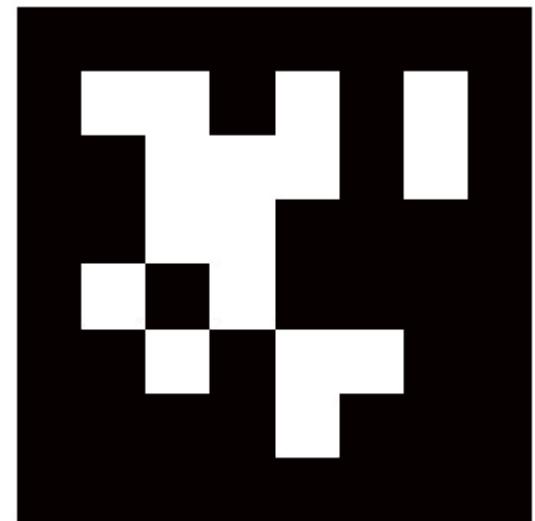
Figura 3.3. Resoluciones estándar.

VISIÓN POR COMPUTADORA



Figura 3.4. Localización y posición con QR en Matlab.

- Existen diversas soluciones mediante librerías y programación para la detección de códigos QR.



Tag36h11

Figura 3.5. April Tag

VISIÓN POR COMPUTADORA

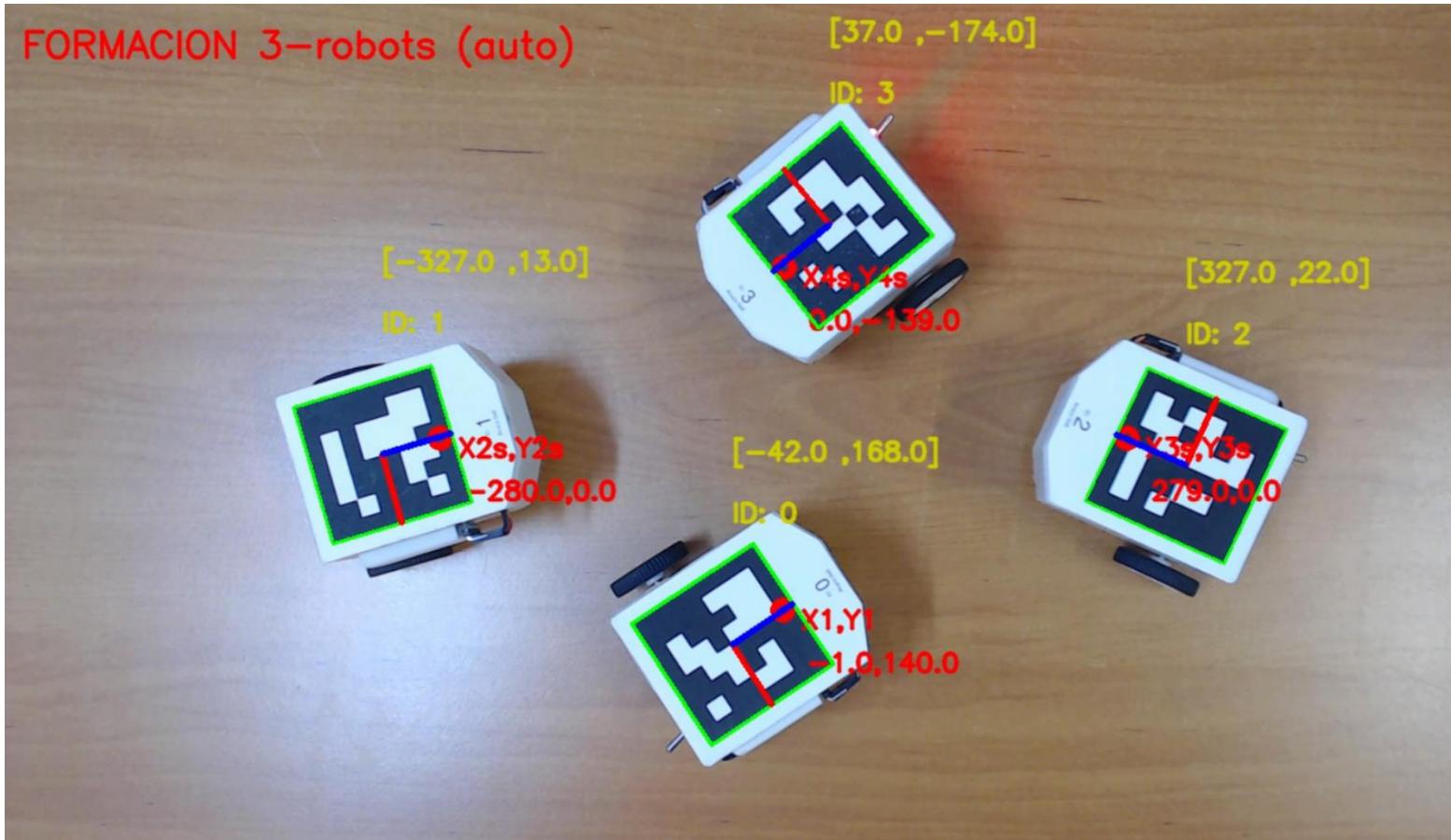


Figura 3.6. Localización y posición con QR en Python.

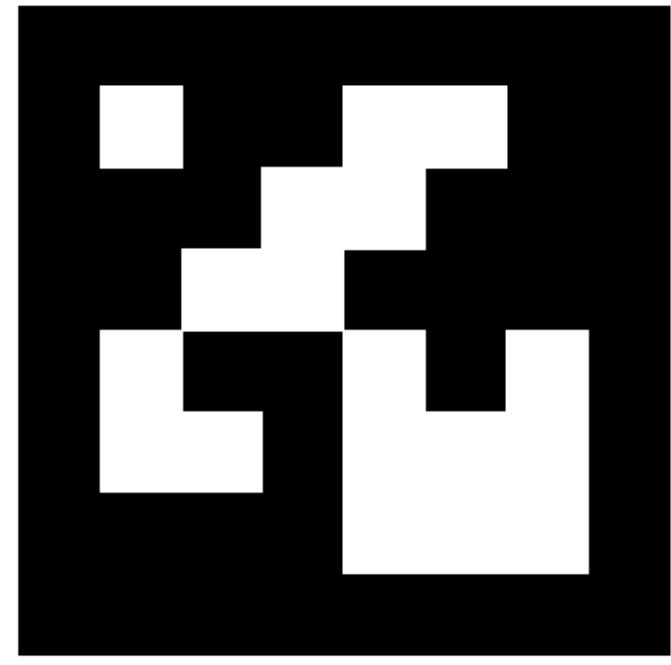


Figura 3.7. Aruco

CÓDIGO PYTHON

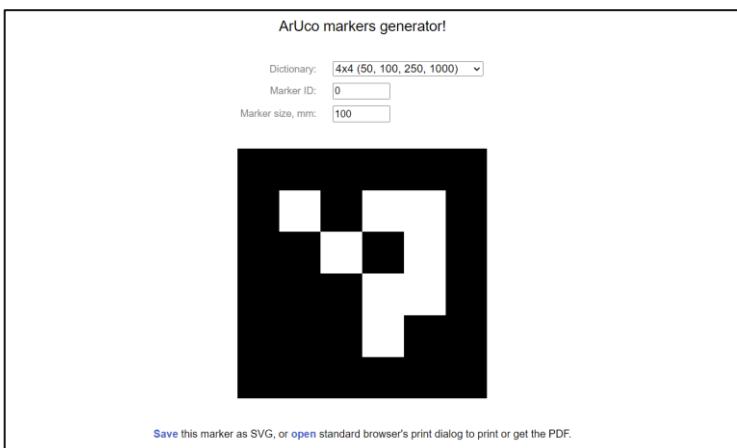
- Descarga el código de ejemplo ***Camera.py*** del repositorio que está en la carpeta ***4_Python_QR_aruco*** y ejecútalo, te permitirá detectar códigos de tipo Aruco.



The screenshot shows a GitHub repository page for 'Robot-uniciclo-Ibero-2.0'. The repository is public and has 1 branch and 0 tags. The commit history shows three commits by 'Hgnto qr' from 441bfe0. The commits are: '1_Solidworks' (solidworks), '2_Impresion 3D' (solidworks), and '3_Codigo Arduino' (solidworks). There are also two initial commits: 'LICENSE' and 'README.md'. The repository has 0 stars, 1 watching, and 0 forks. The 'About' section includes a description of the project: 'Robot unico para prácticas de control y coordinación de robots móviles, para plataforma de bajo costo mediante uso de python, cámara usb y computadora personal. Universidad Iberoamericana Puebla.' It also lists 'Readme', 'GPL-3.0 license', 'Activity', '0 stars', '1 watching', and '0 forks'.

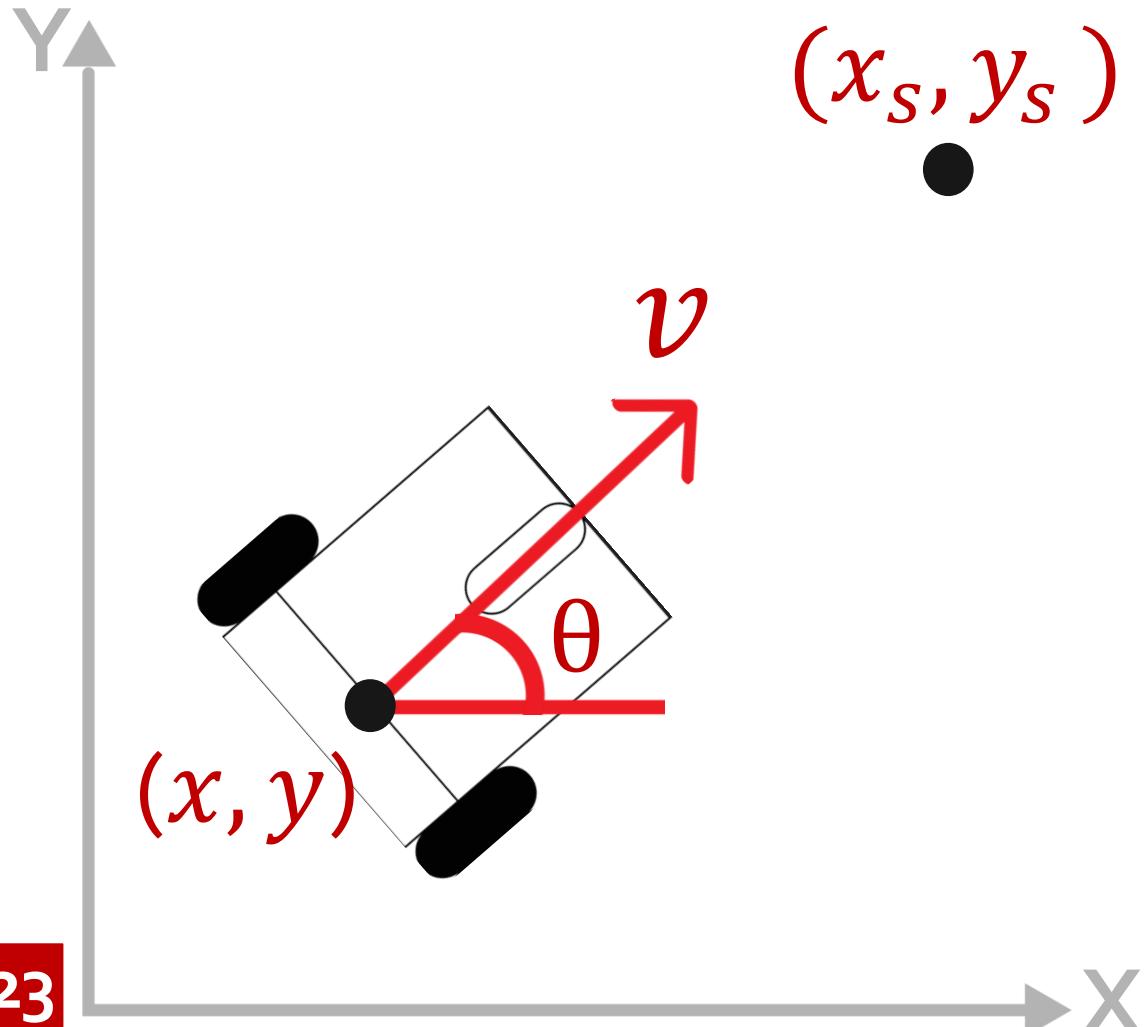
- Es necesario tener instalado las librerías **OpenCV**, **Numpy** y **Math** de python.

```
1 import cv2 #pip install opencv-python, opencv-contrib-python
2 import numpy as np #pip install numpy
3 import math #default pip install python-math
```



4. ESTRATEGIA DE CONTROL: PARTÍCULA

CONVERGENCIA



(x, y) → posición actual
 θ → orientación

(x_s, y_s) → posición deseada
(star point)

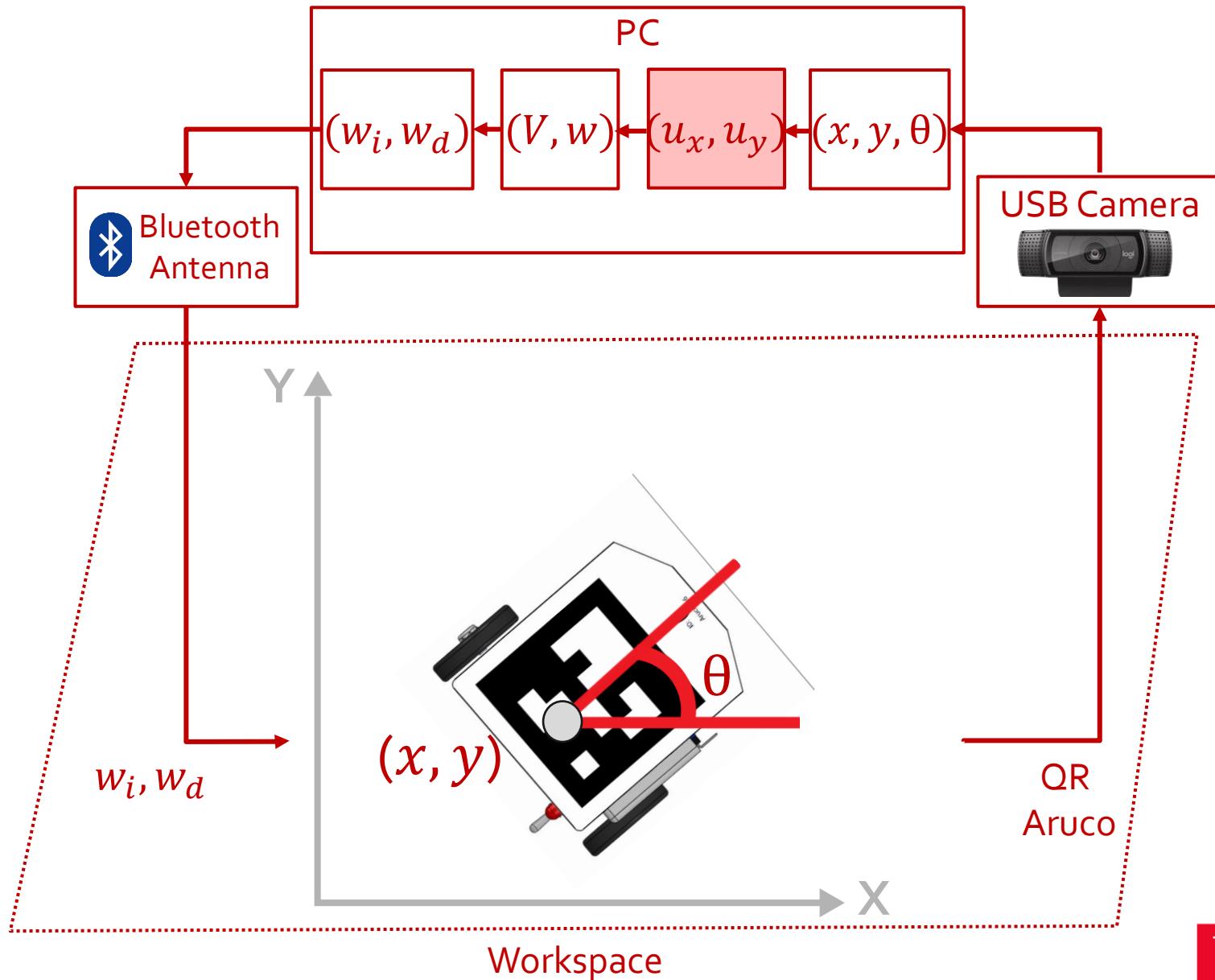
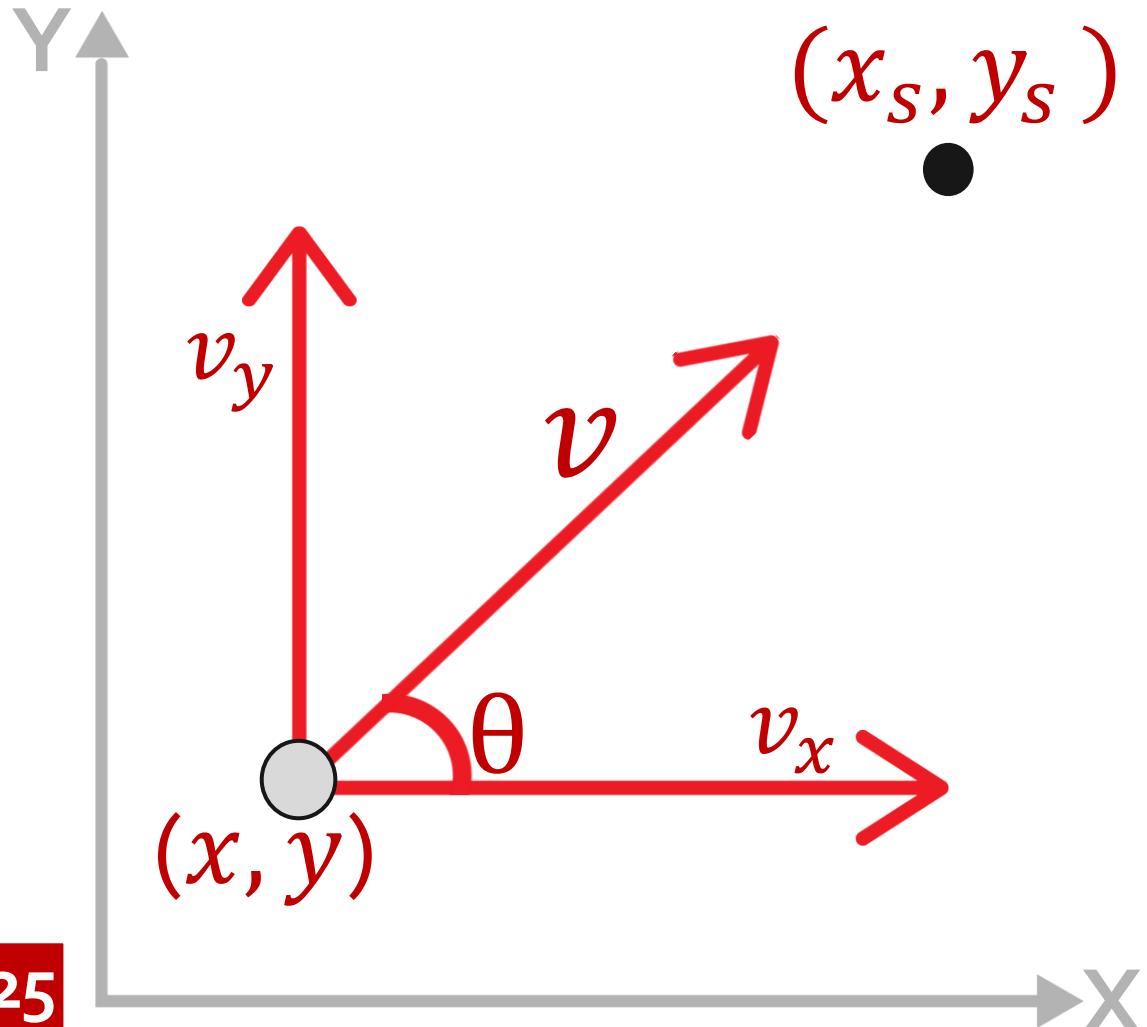


Figura 4.o. Plataforma de experimentación.

CONVERGENCIA

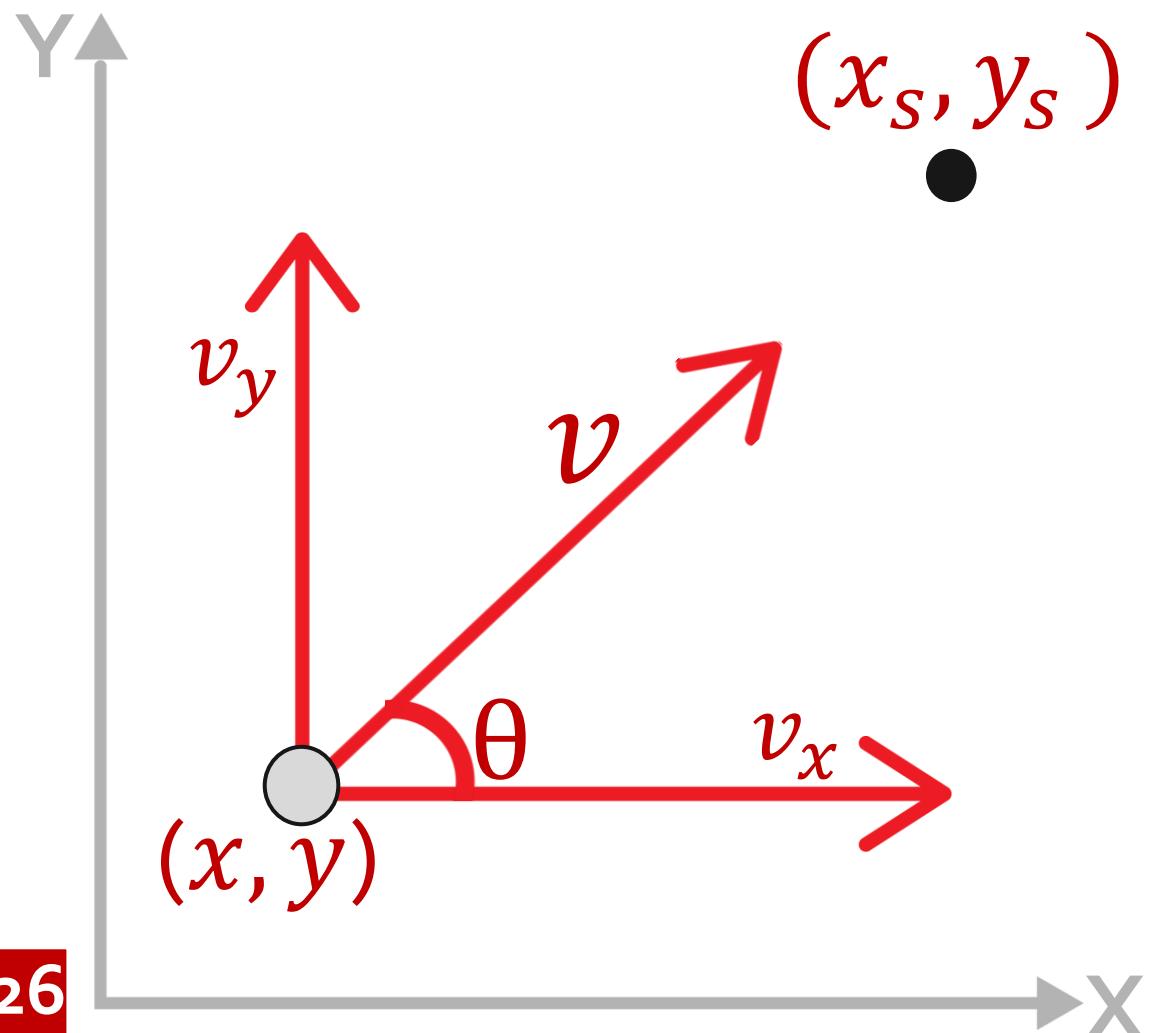


$(x, y) \rightarrow$ posición actual
 $\theta \rightarrow$ orientación

(x_s, y_s) \rightarrow posición deseada
(star point)

$$\begin{aligned}v &\rightarrow \text{Velocidad Lineal} \\v_x &= v \cos \theta \\v_y &= v \sin \theta\end{aligned}$$

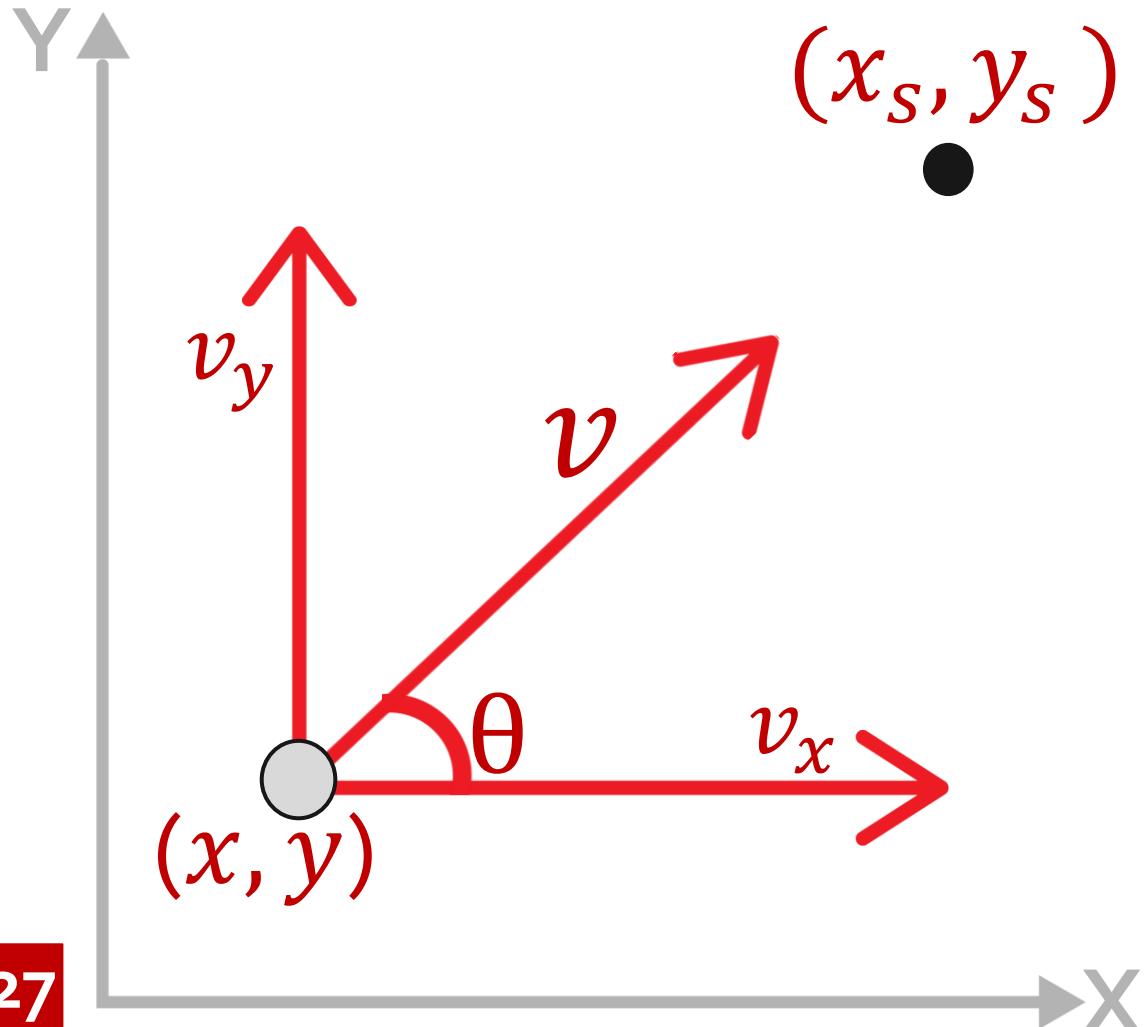
CONVERGENCIA



$(x, y) \rightarrow$ posición actual
 $(x_s, y_s) \rightarrow$ posición deseada
(star point)

$$\begin{aligned} k &= 1 \\ error_x &= x - x_s \\ error_y &= y - y_s \\ u_x &= -k * error_x \\ u_y &= -k * error_y \end{aligned}$$

CONVERGENCIA



$$k = 1$$

$$\text{error}_x = x - x_s$$

$$\text{error}_y = y - y_s$$

$$u_x = -k * \text{error}_x$$

$$u_y = -k * \text{error}_y$$

$$x_{(t+1)} = x_{(t)} + u_x * t$$

$$y_{(t+1)} = y_{(t)} + u_y * t$$

CONVERGENCIA

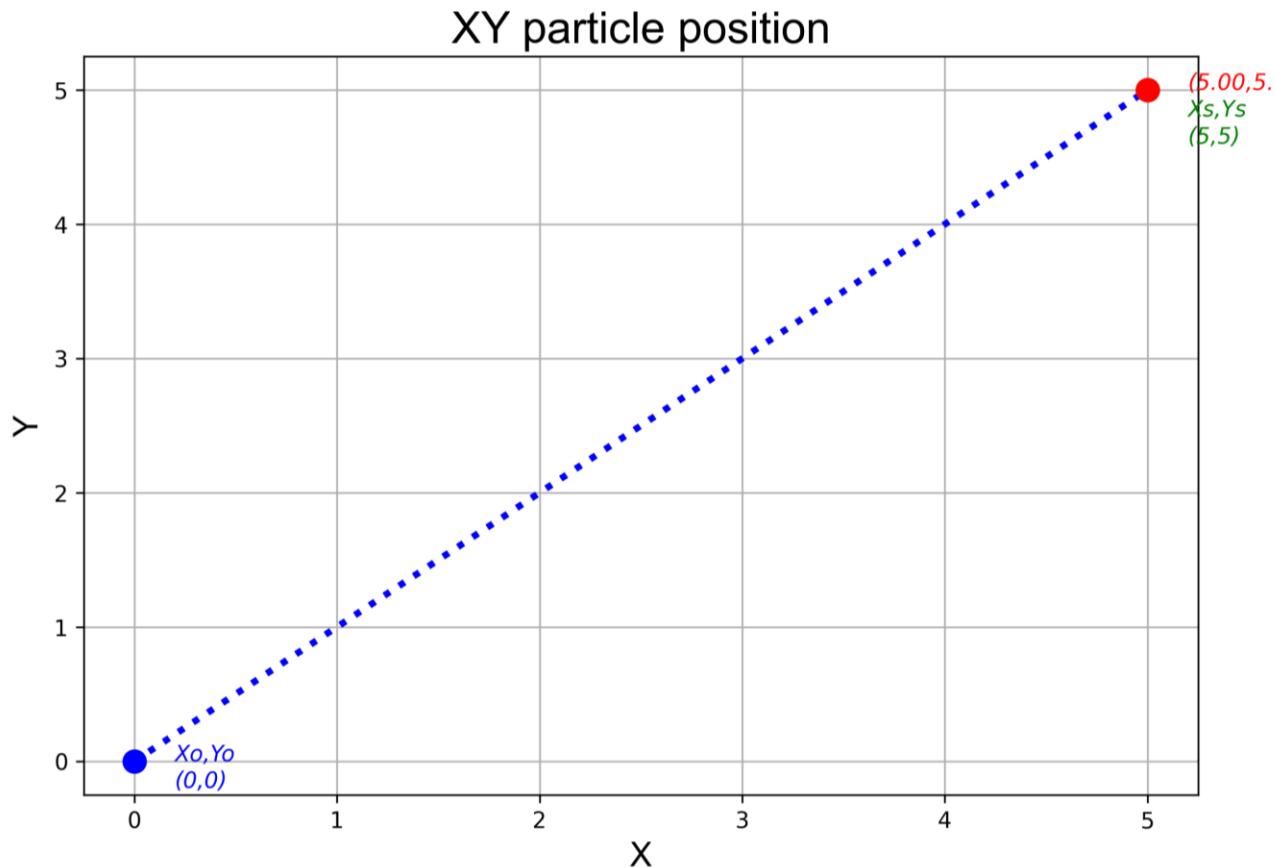


Figura 4.1. Grafica posición XY.

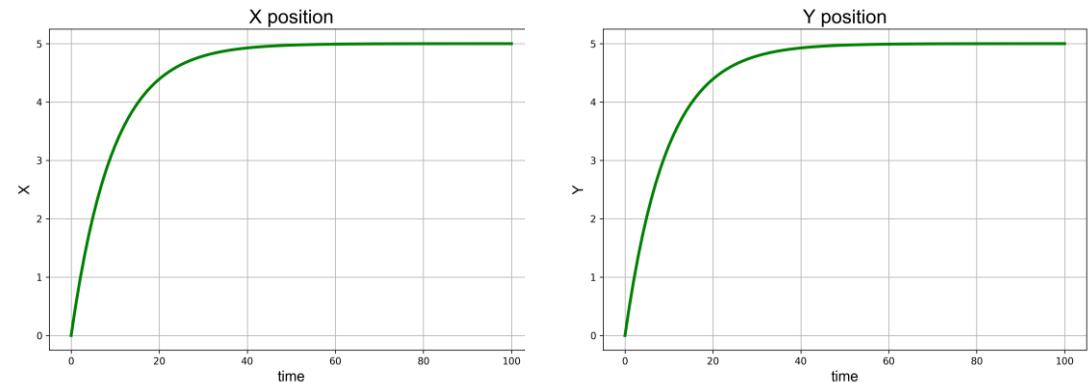


Figura 4.2. Graficas de posición.

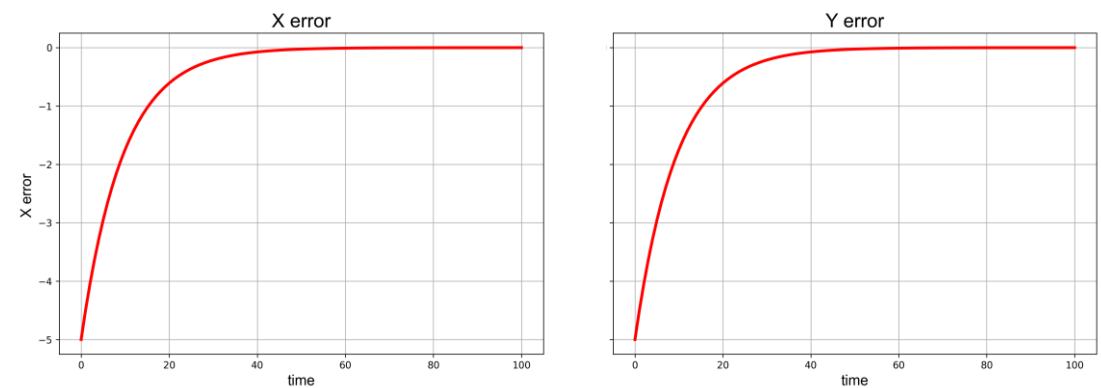
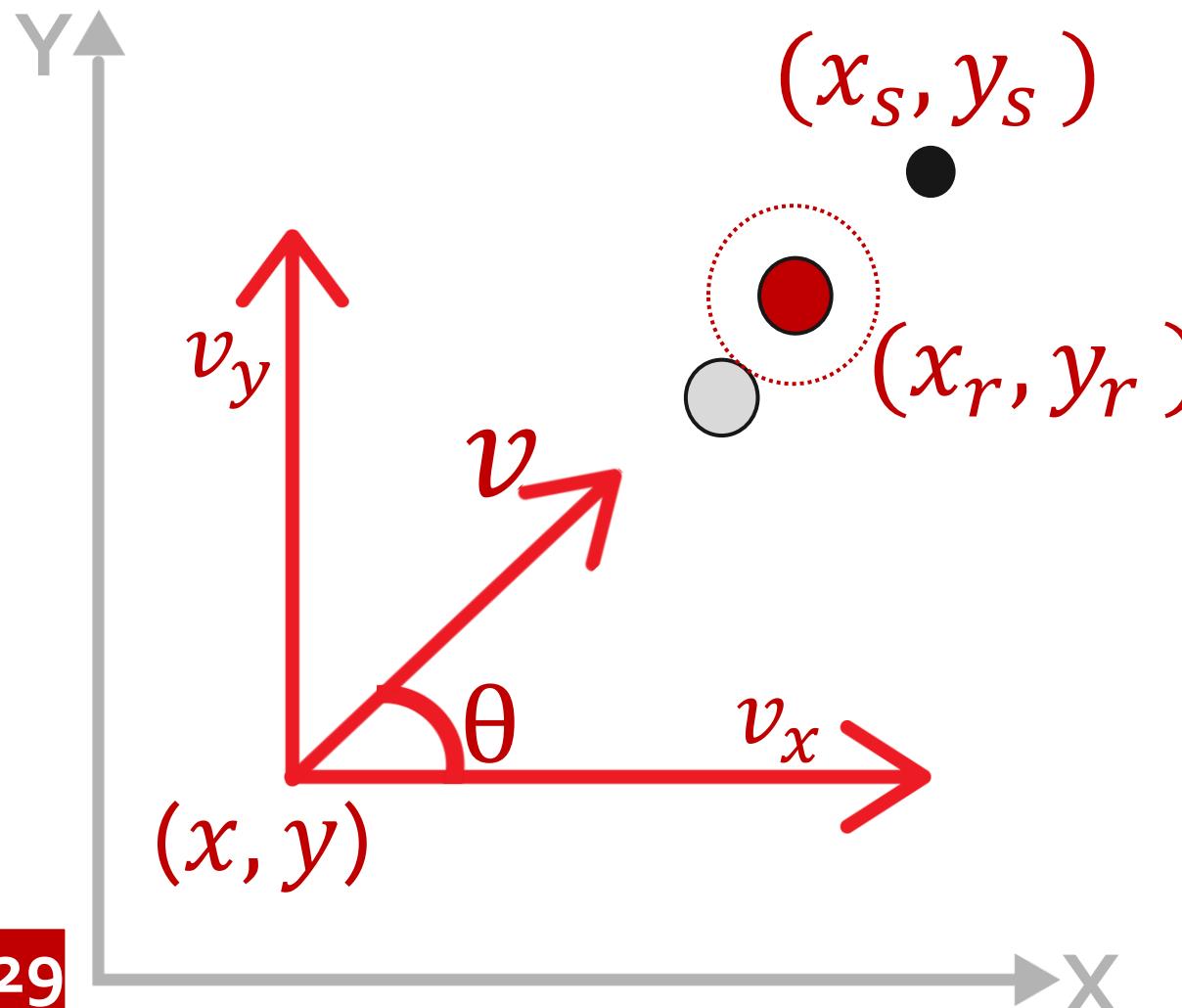


Figura 4.3. Graficas de error.

EVITAR PARTICULARA

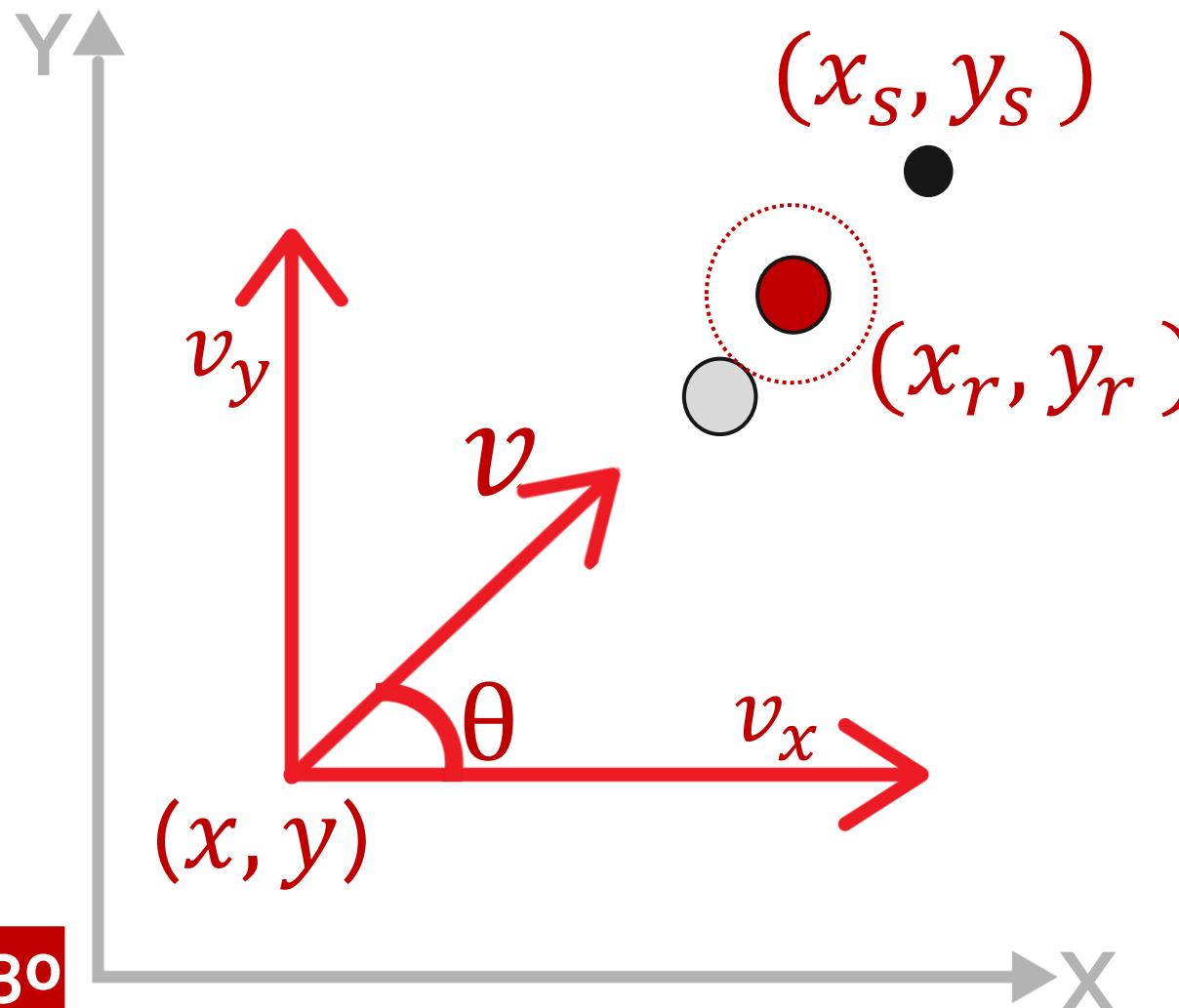


$(x, y) \rightarrow$ posición actual
 $(x_s, y_s) \rightarrow$ posición deseada
 $(x_r, y_r) \rightarrow$ posición evitar

$a_x, a_y \rightarrow$ Potencial atractivo
 $R_x, R_y \rightarrow$ Potencial repulsivo

$$u_x = a_x + R_x$$
$$u_y = a_y + R_y$$

EVITAR PARTICULA



$$u_x = a_x + R_x$$
$$u_y = a_y + R_y$$

$a_x, a_y \rightarrow$ Potencial atractivo

$$k = 1$$

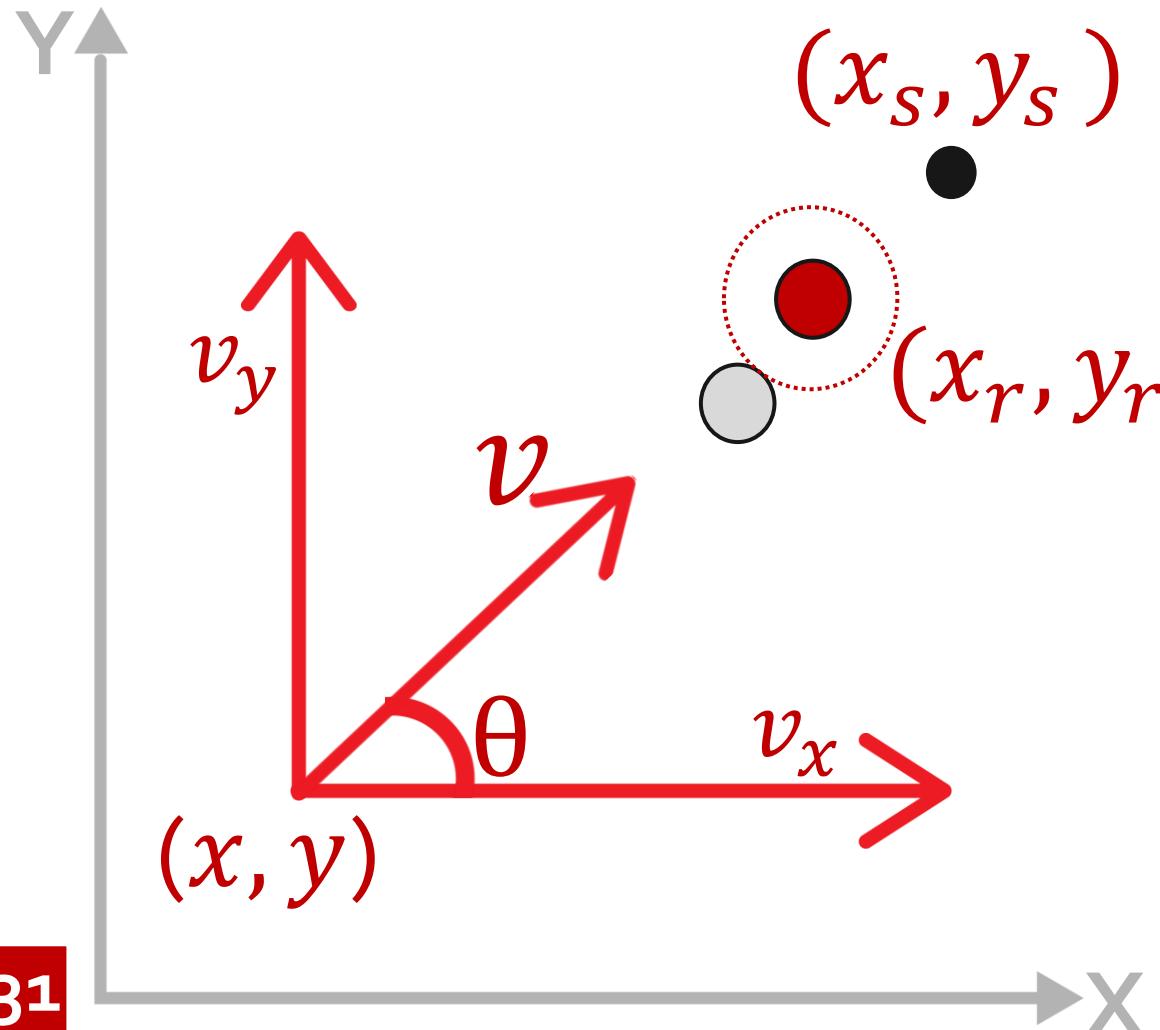
$$\text{error}_x = x - x_s$$

$$\text{error}_y = y - y_s$$

$$a_x = -k * \text{error}_x$$

$$a_y = -k * \text{error}_y$$

EVITAR PARTICULA



$$u_x = a_x + R_x$$
$$u_y = a_y + R_y$$

$R_x, R_y \rightarrow$ Potencial repulsivo

$$r=2$$

$$b = (x - x_r)^2 + (y - y_r)^2$$

$$G_x = 2 * \left(\left(\frac{1}{b} \right) - \left(\frac{1}{r^2} \right) \right) * \frac{db(x)}{dx}$$

$$\frac{db(x)}{dx} = -2 * (x - x_r) * \left(\frac{1}{b} \right)^2$$

$$\text{et } et = 0.01 \quad R_x = -et * GR_x$$

EVITAR PARTICULA

$$r = 2$$

$$b = (x - x_r)^2 + (y - y_r)^2$$

$$GR_x = 2 * \left(\left(\frac{1}{b} \right) - \left(\frac{1}{r^2} \right) \right) * \frac{db(x)}{dx}$$

$$\frac{db(x)}{dx} = -2 * (x - x_r) * \left(\frac{1}{b} \right)^2$$

$$et = 0.01 \quad R_x = -et * GR_x$$

$R_x, R_y \rightarrow$ Potencial repulsivo

$$r = 2$$

$$b = (x - x_r)^2 + (y - y_r)^2$$

$$GR_y = 2 * \left(\left(\frac{1}{b} \right) - \left(\frac{1}{r^2} \right) \right) * \frac{db(y)}{dy}$$

$$\frac{db(y)}{dy} = -2 * (y - y_r) * \left(\frac{1}{b} \right)^2$$

$$et = 0.01 \quad R_y = -et * GR_y$$

EVITAR PARTICULARA

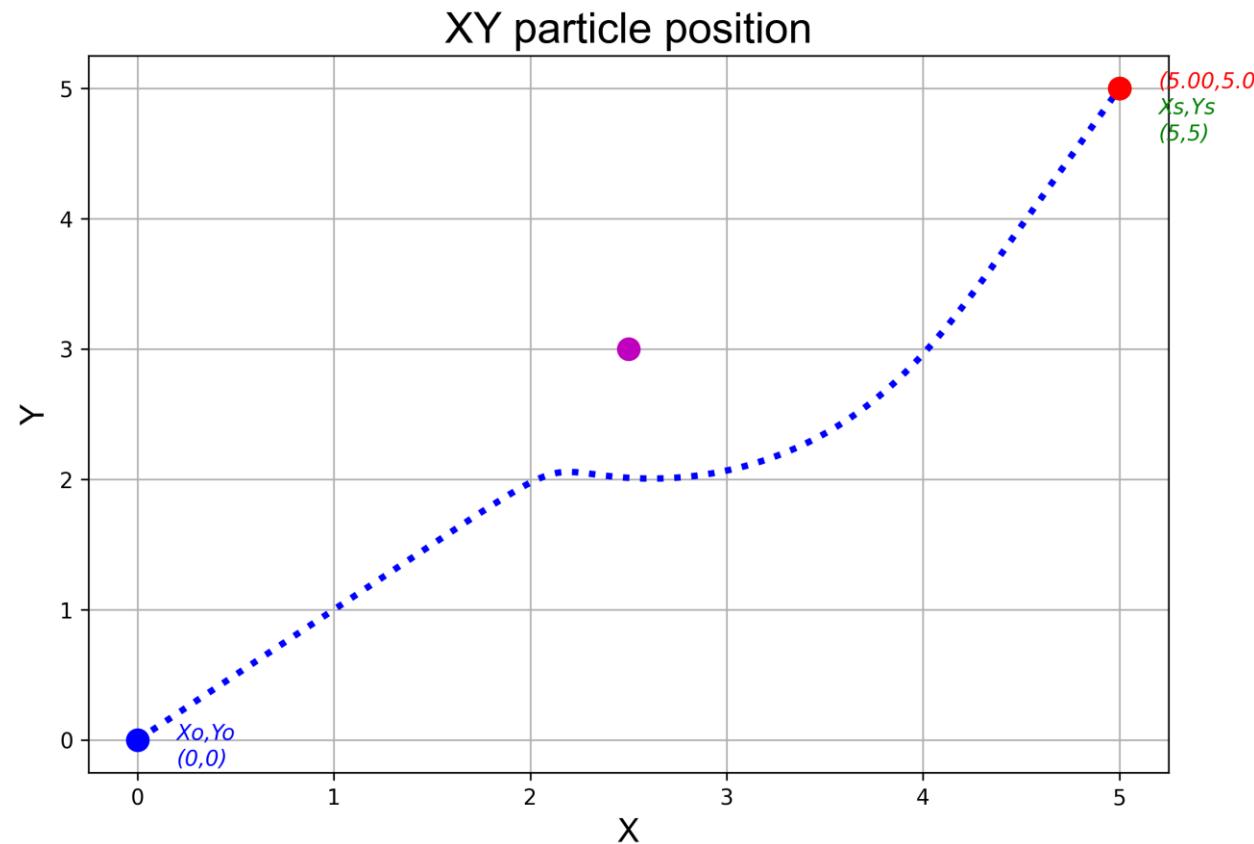


Figura 4.4. Grafica posición XY.

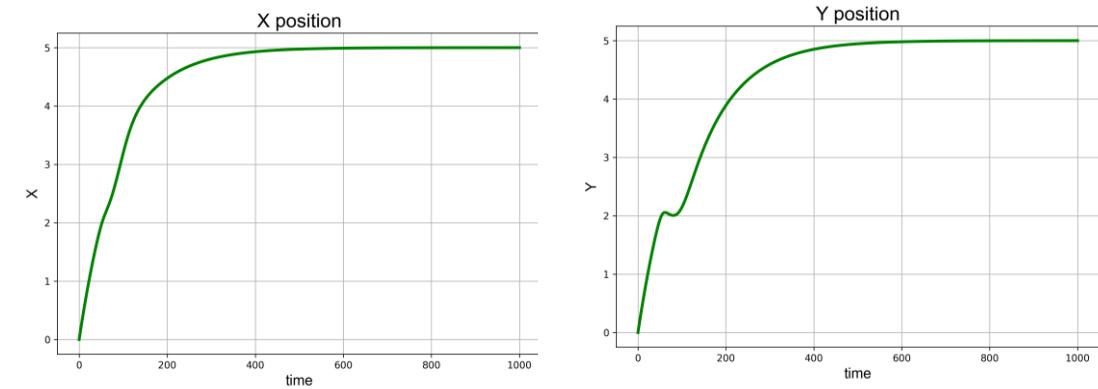


Figura 4.5. Graficas de posición.

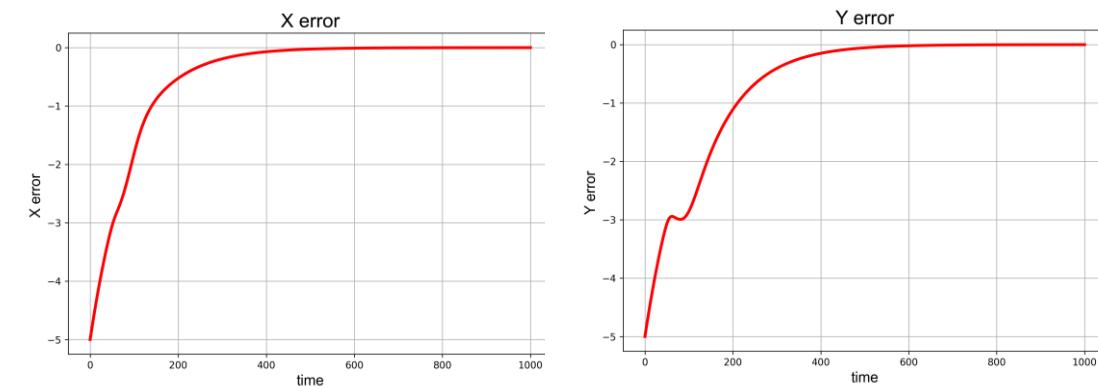


Figura 4.6. Graficas de error.

EVITAR PARTICULARA

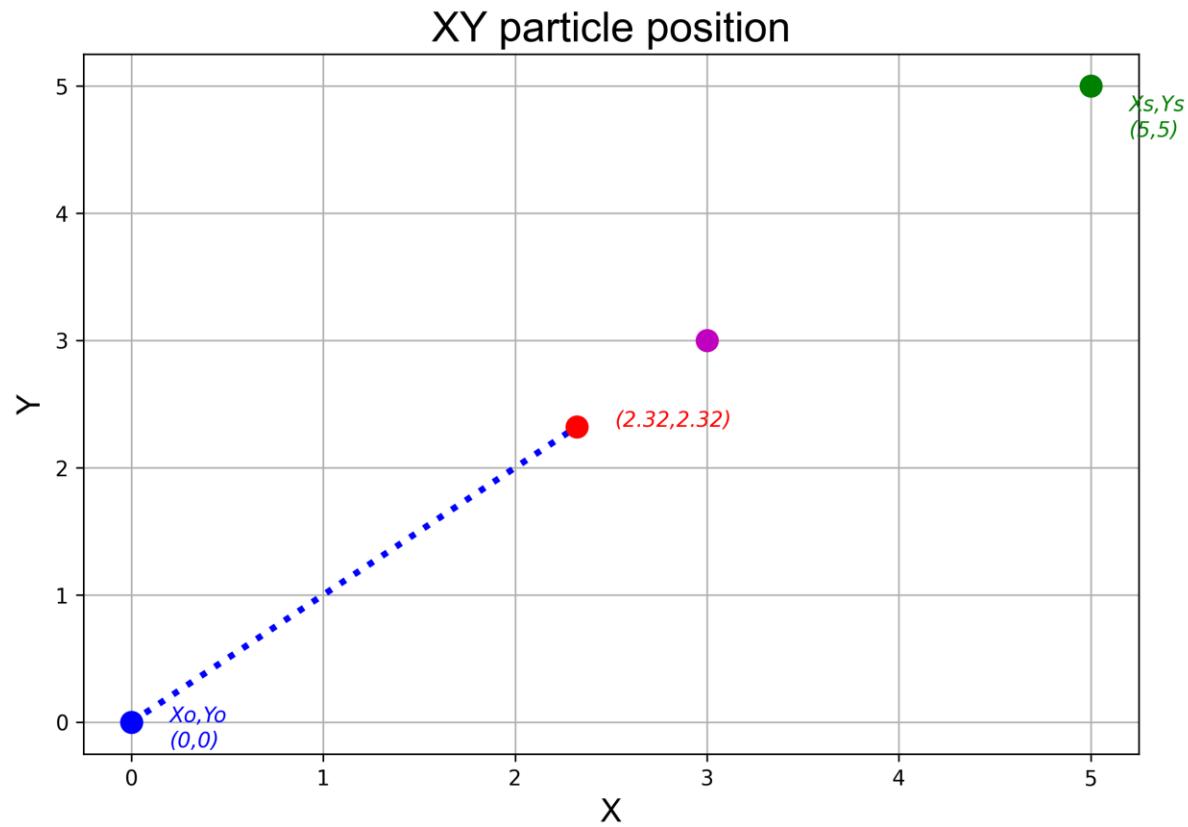


Figura 4.7. Grafica posición XY.

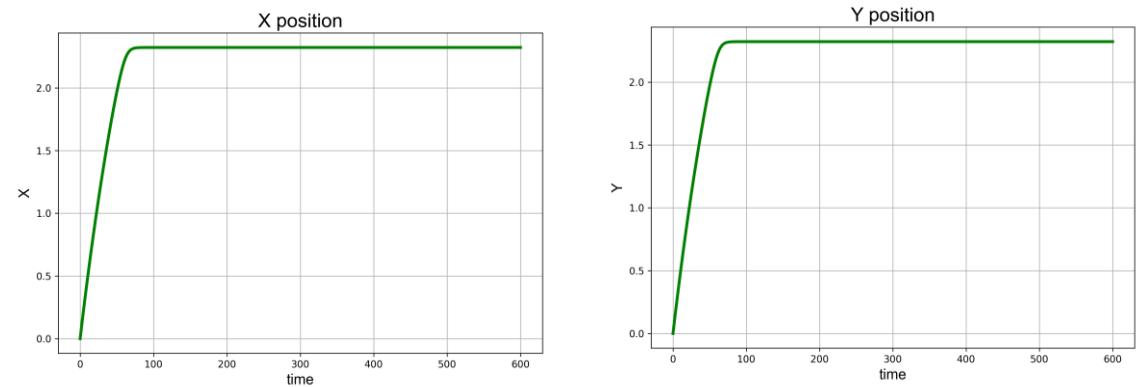


Figura 4.8. Graficas de posición.

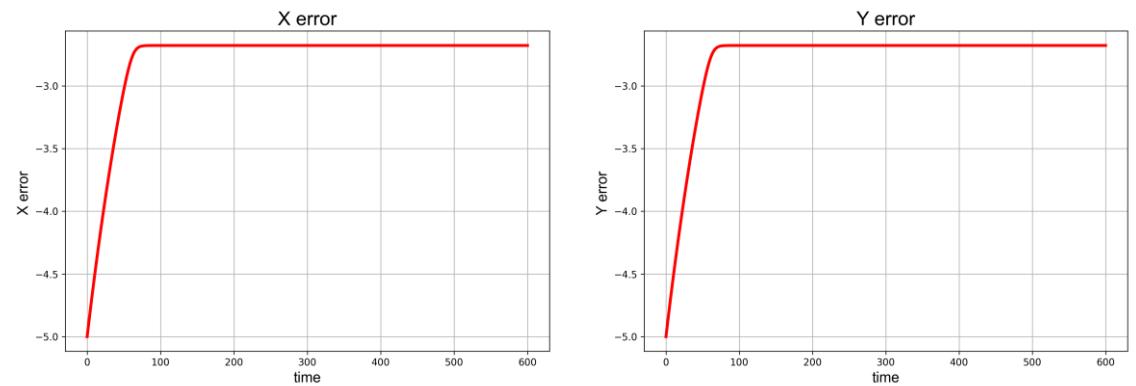
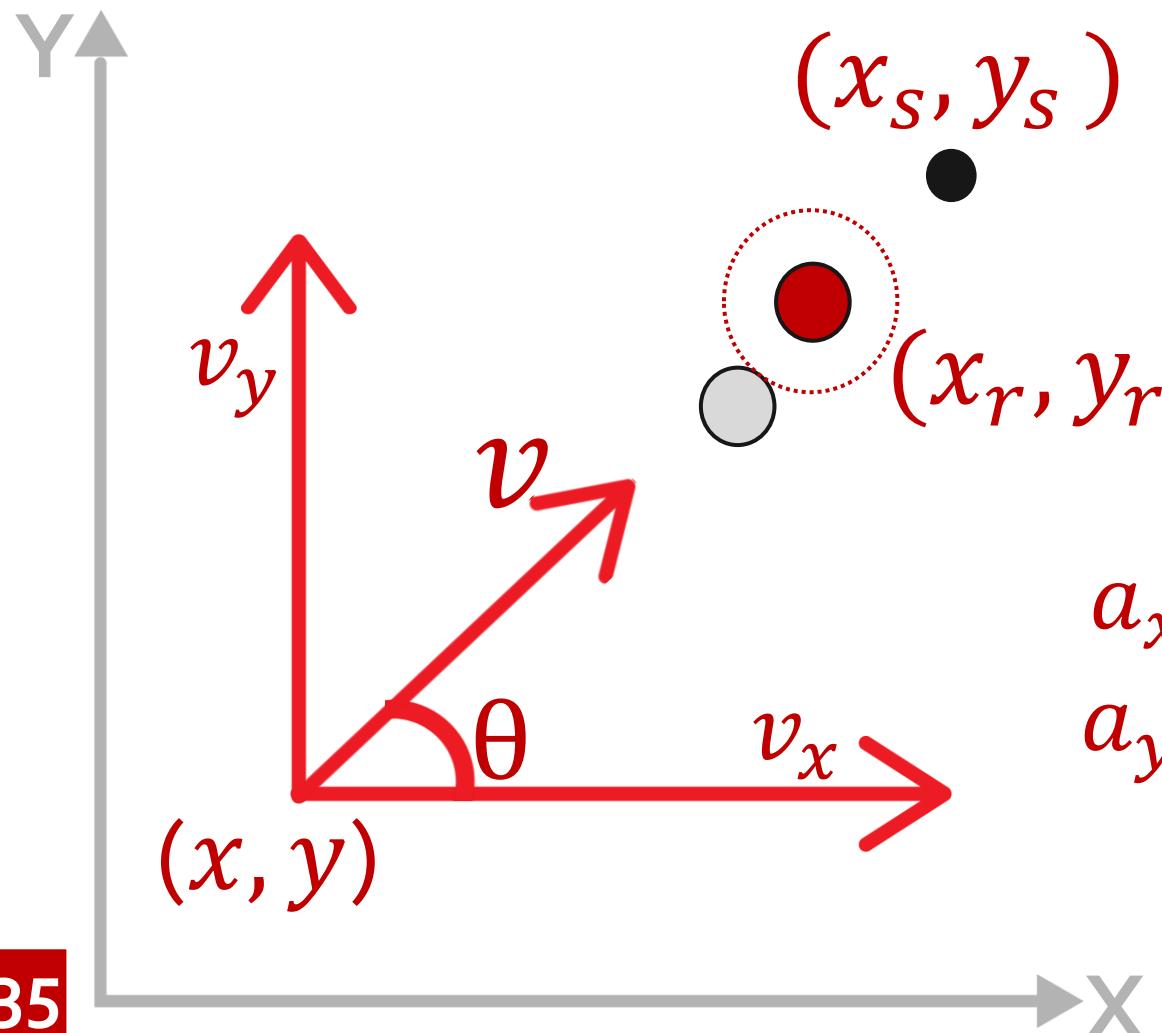


Figura 4.9. Graficas de error.

EVITAR PARTICULARA



$$u_x = a_x + R_x$$
$$u_y = a_y + R_y$$

$a_x, a_y \rightarrow$ Potencial atractivo en espiral

$$k = 1 \quad c = 0.01$$
$$\text{error}_x = x - x_s$$

$$\text{error}_y = y - y_s$$

$$a_x = (-k * \text{error}_x) - (c * \text{error}_y)$$
$$a_y = (-k * \text{error}_y) - (c * \text{error}_x)$$

EVITAR PARTICULARA

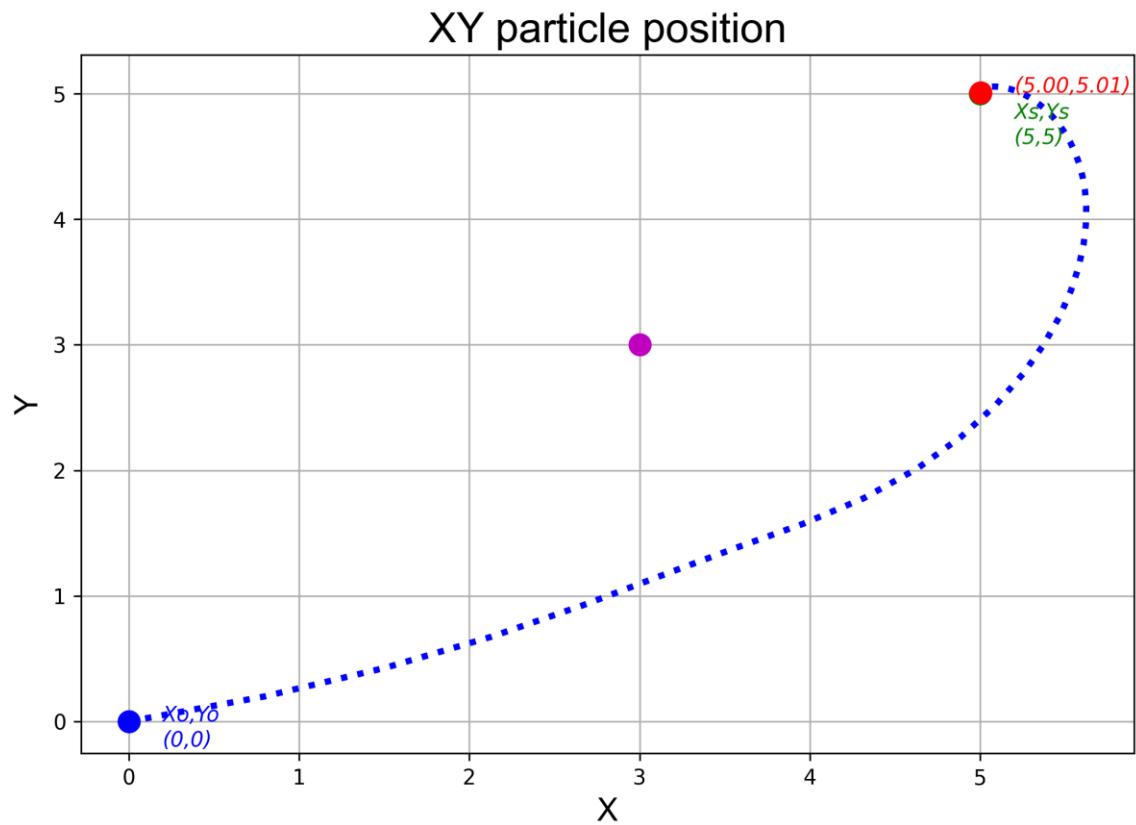


Figura 4.10. Grafica posición XY.

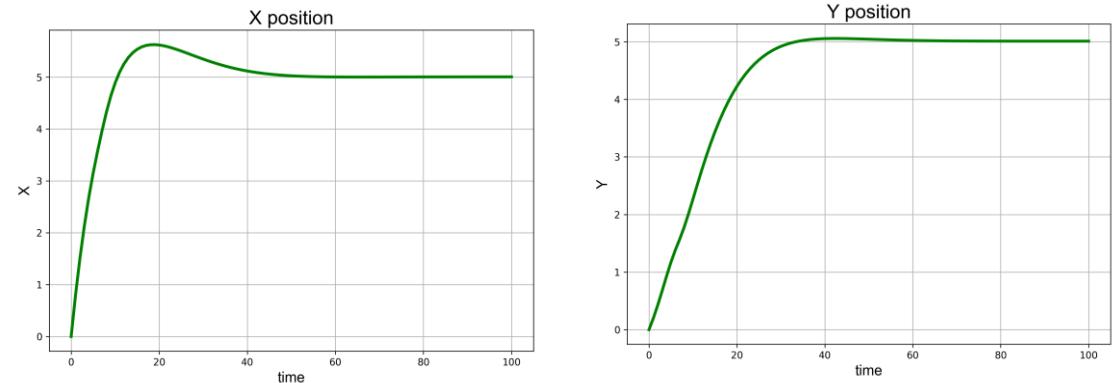


Figura 4.11. Graficas de posición.

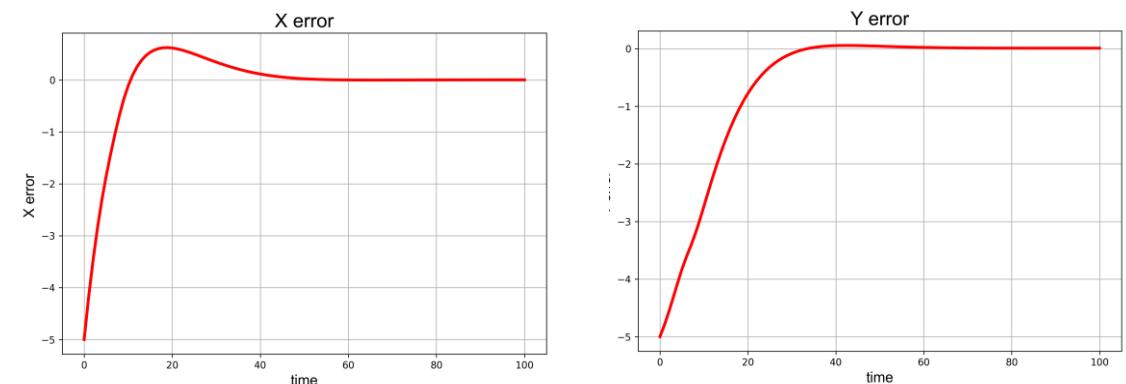


Figura 4.12. Graficas de error.

EVITAR PARTICULARA

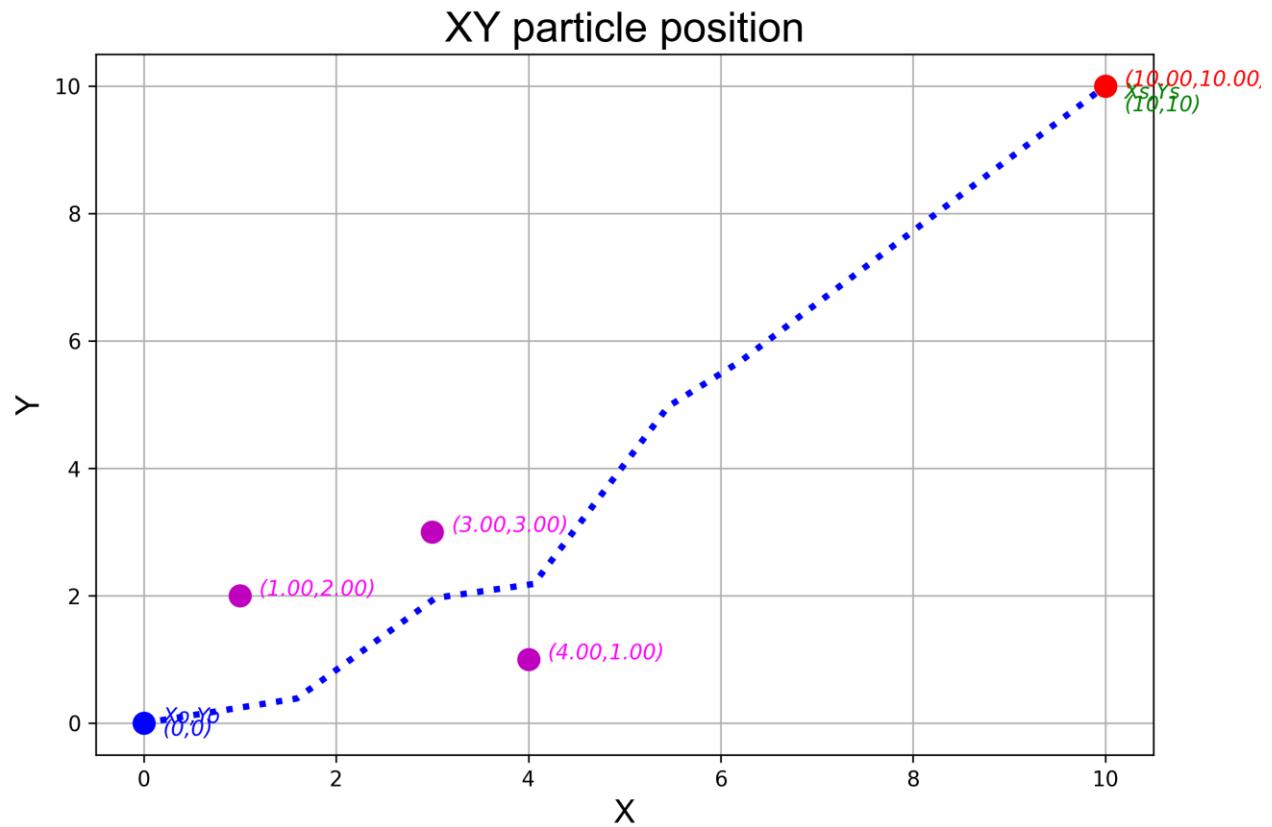


Figura 4.13. Grafica posición XY.

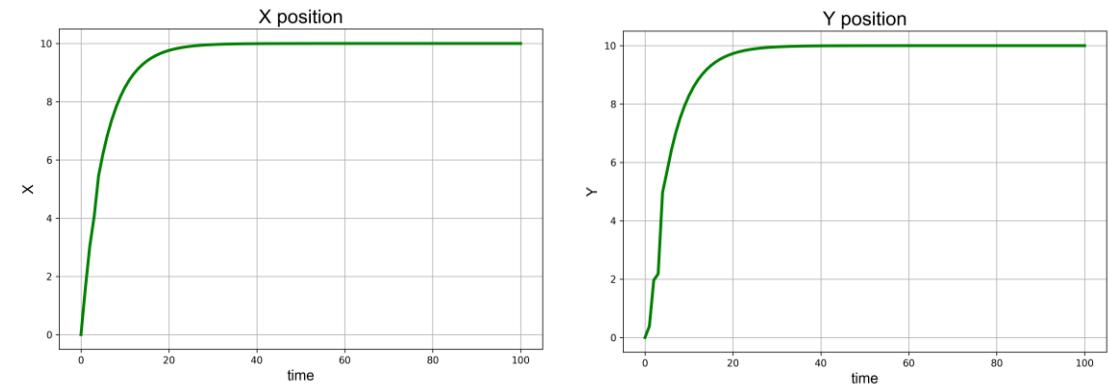


Figura 4.14. Graficas de posición.

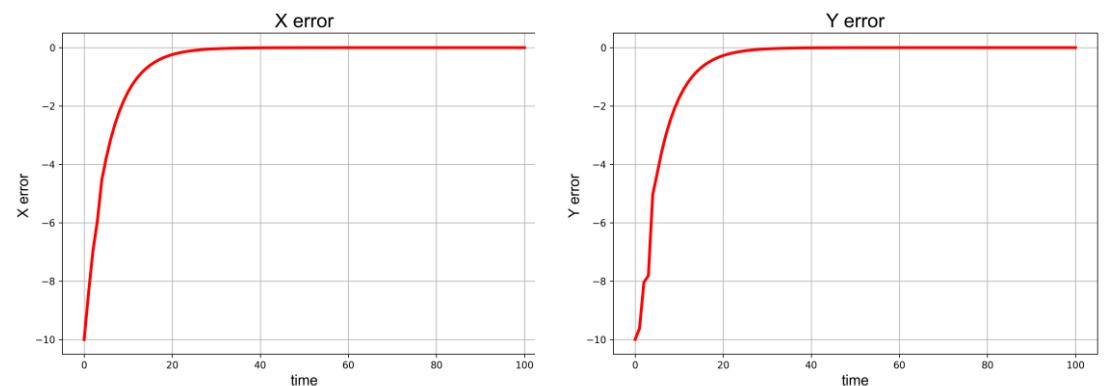


Figura 4.15. Graficas de error.

EVITAR PARTICULA

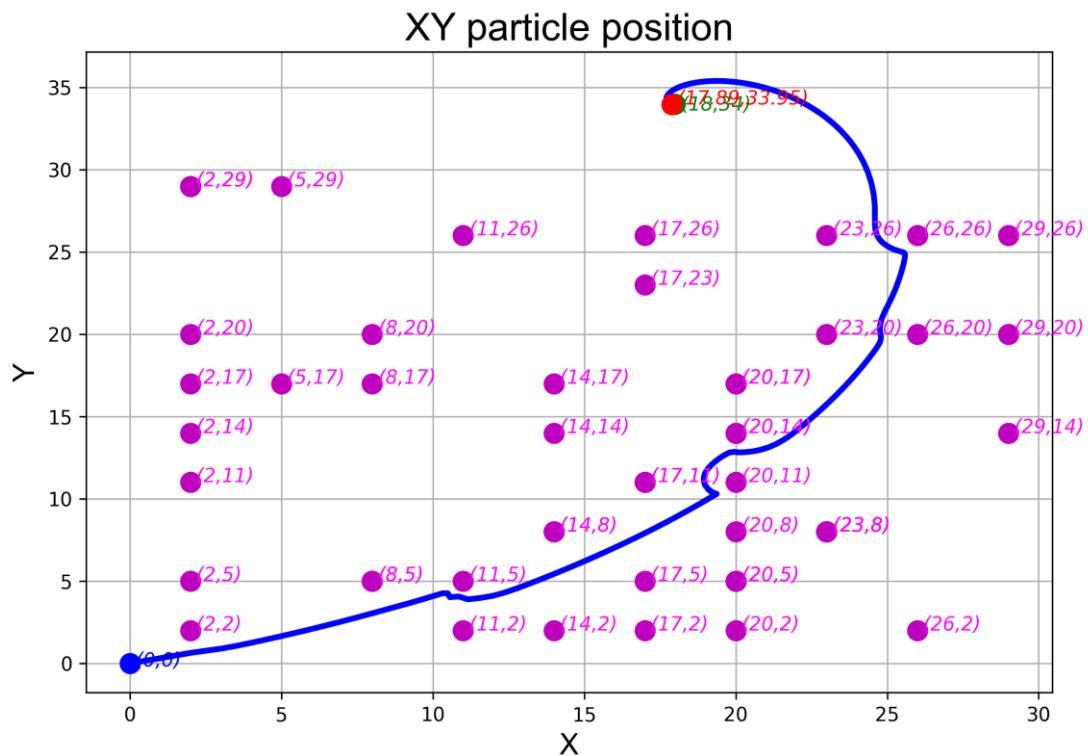


Figura 4.16. Grafica posición XY.

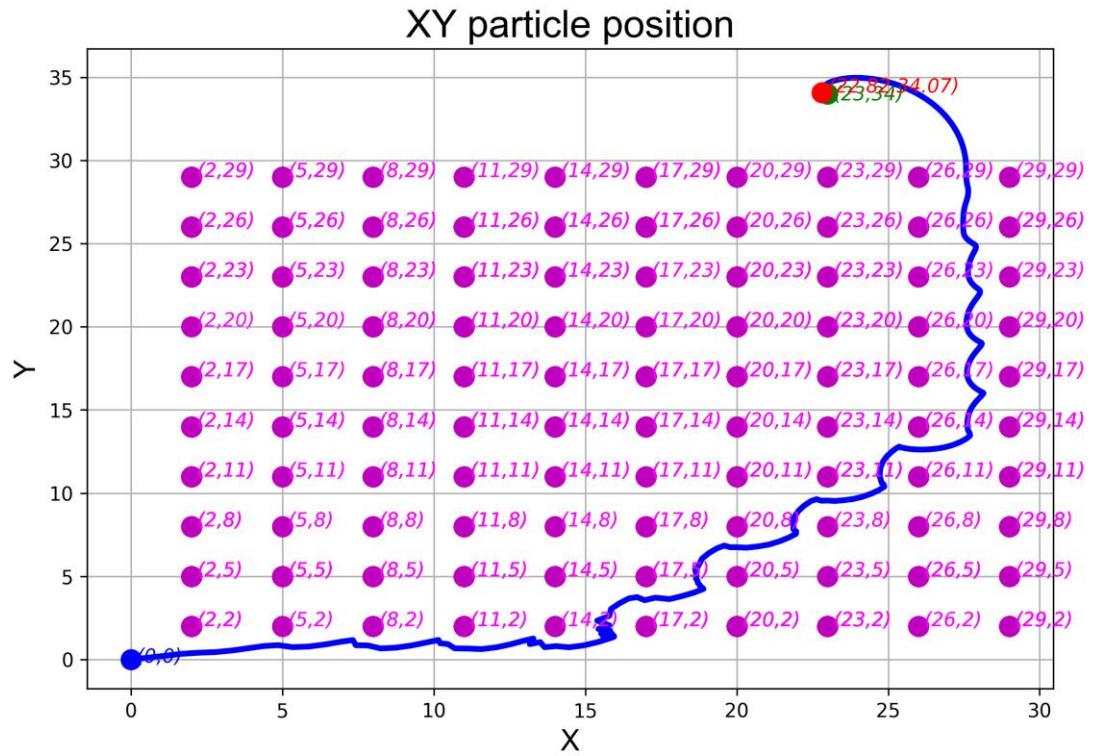
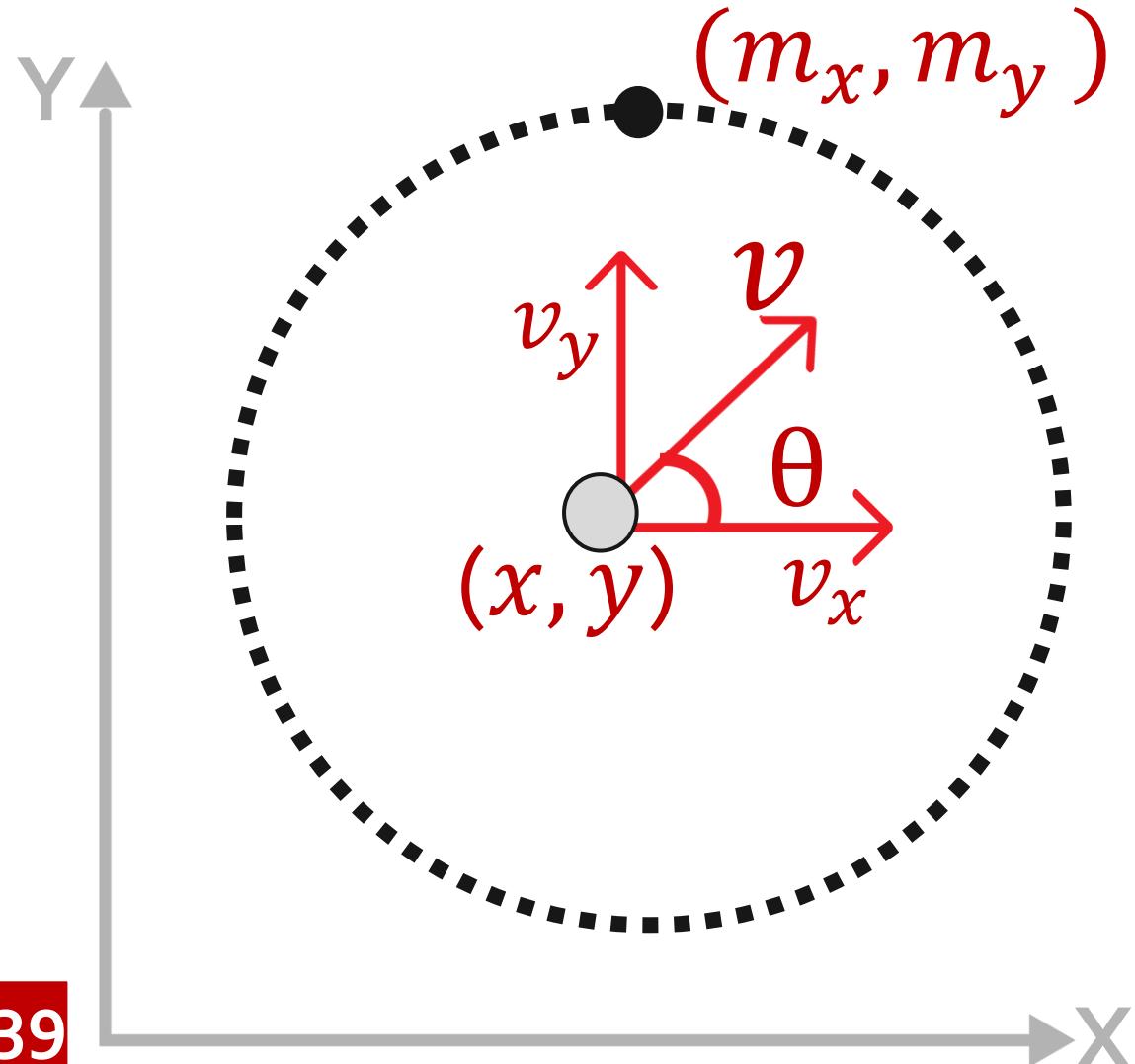


Figura 4.17. Grafica posición XY.

TRAYECTORIA



$(x, y) \rightarrow$ posición actual

$\theta \rightarrow$ orientación

$m_x(t), m_y(t) \rightarrow$ Trayectoria

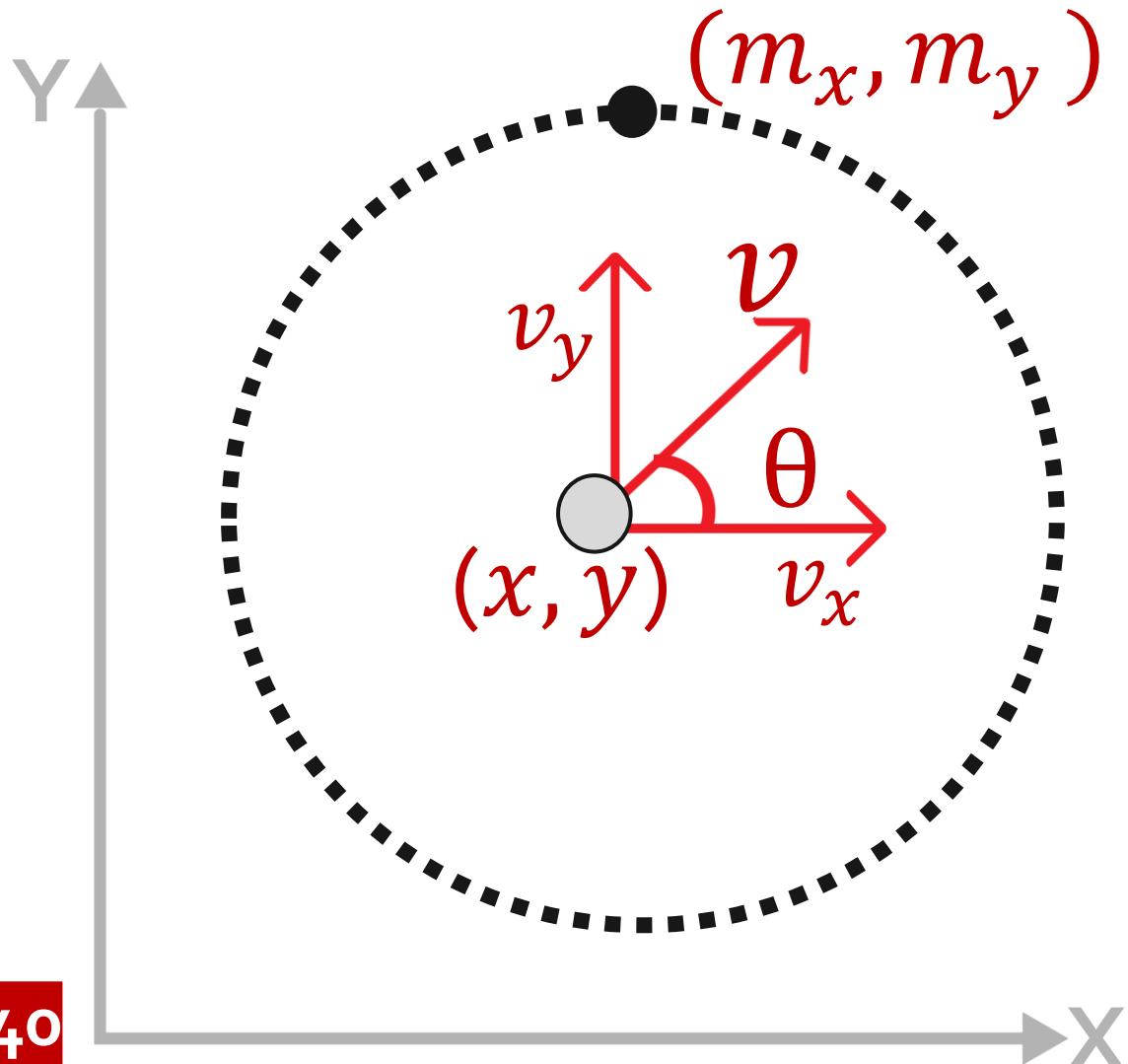
$cx = 0, cy = 0, r = 3$

$$T = 100, w = \frac{2\pi}{T}$$

$$m_x = cx + r * \cos(wt)$$

$$m_y = cy + r * \sin(wt)$$

TRAYECTORIA



$m_x(t), m_y(t) \rightarrow$ Trayectoria

$$k = 1$$

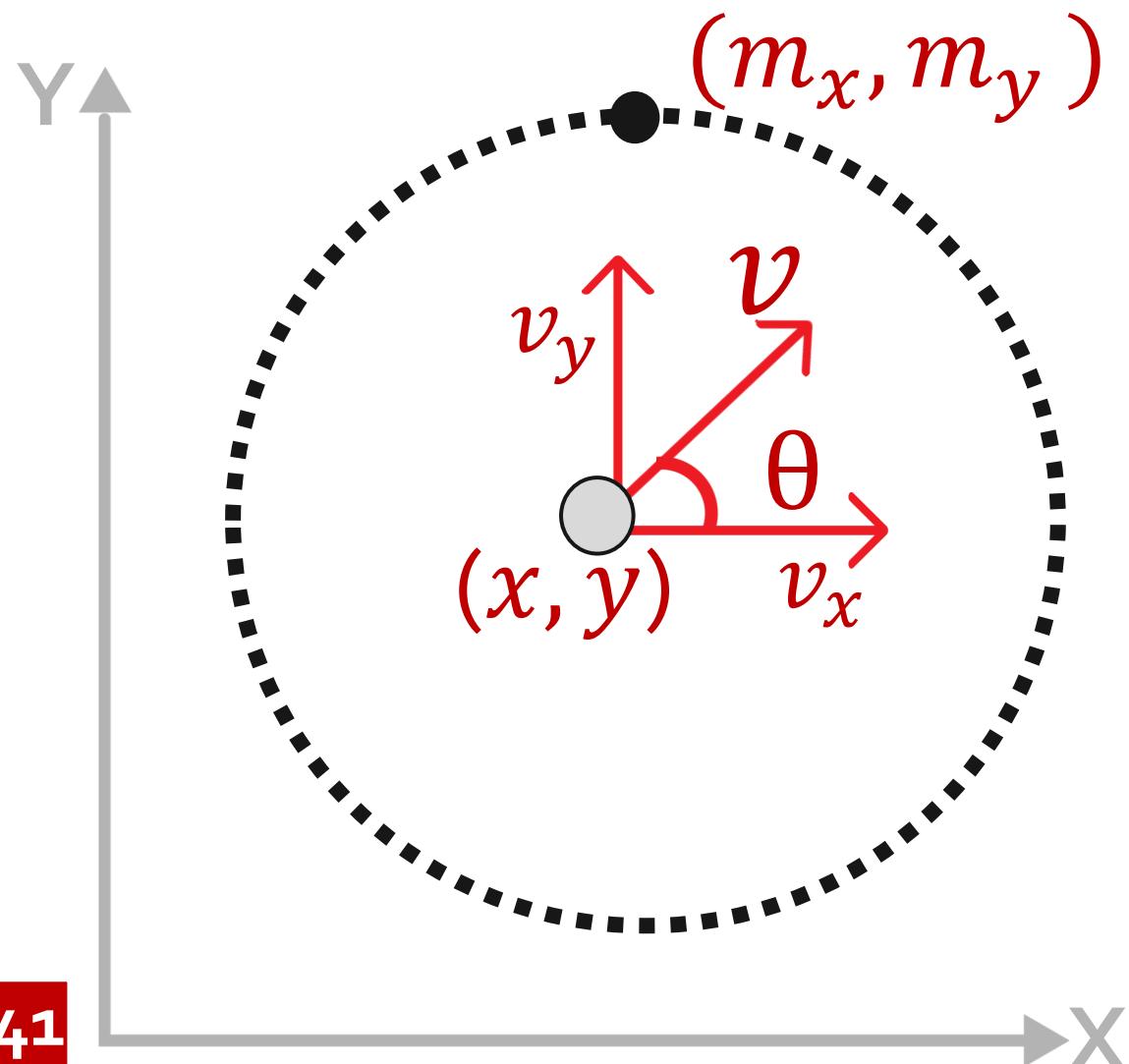
$$\text{error}_x = x - m_x$$

$$\text{error}_y = y - m_y$$

$$u_x = (-k * \text{error}_x) + \dot{m}_x$$

$$u_y = (-k * \text{error}_y) + \dot{m}_y$$

TRAYECTORIA



$m_x(t), m_y(t) \rightarrow$ Trayectoria

$$cx = 0, cy = 0, r = 3$$

$$T = 100, w = \frac{2\pi}{T}$$

$$m_x = cx + r * \cos(wt)$$

$$m_y = cy + r * \sin(wt)$$

$$\dot{m}_x = -r * w * \sin(wt)$$

$$\dot{m}_y = r * w * \cos(wt)$$

TRAYECTORIA

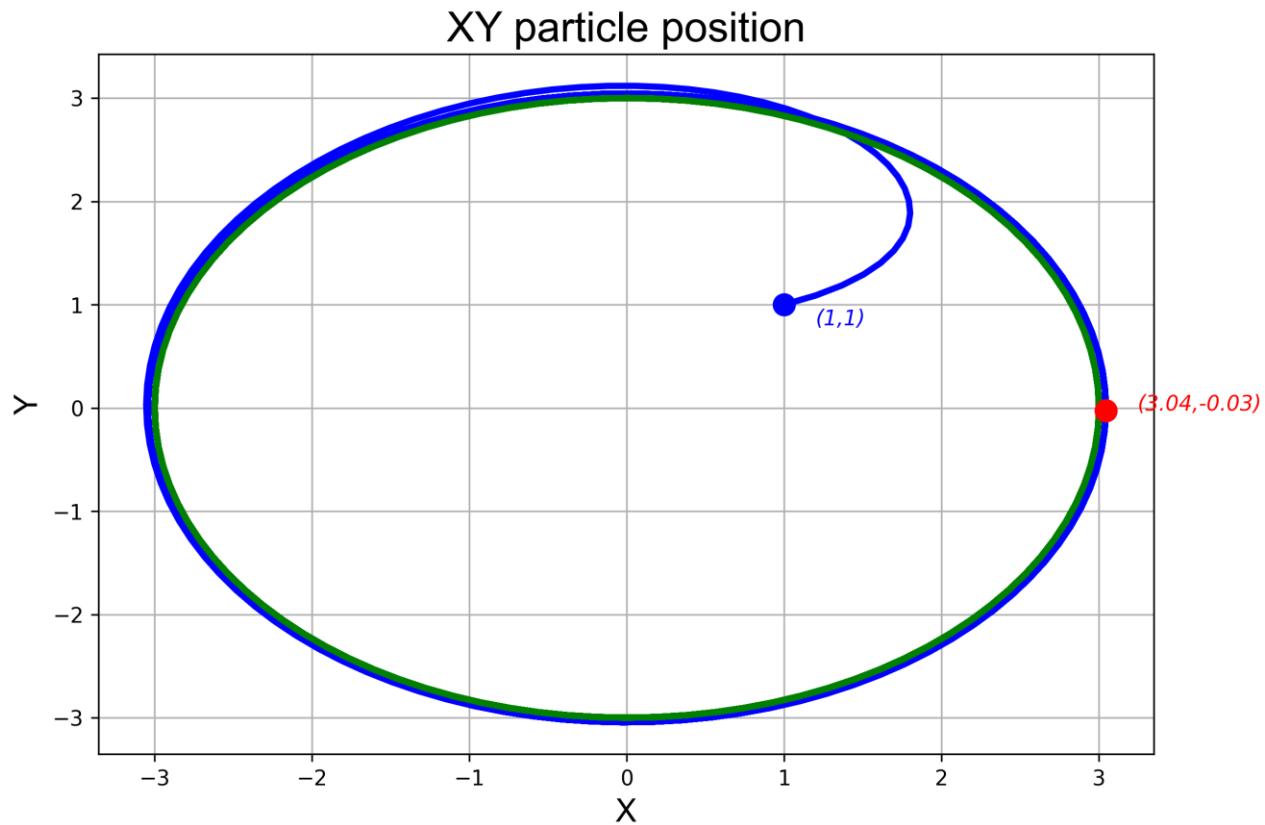


Figura 4.18. Grafica posición XY.

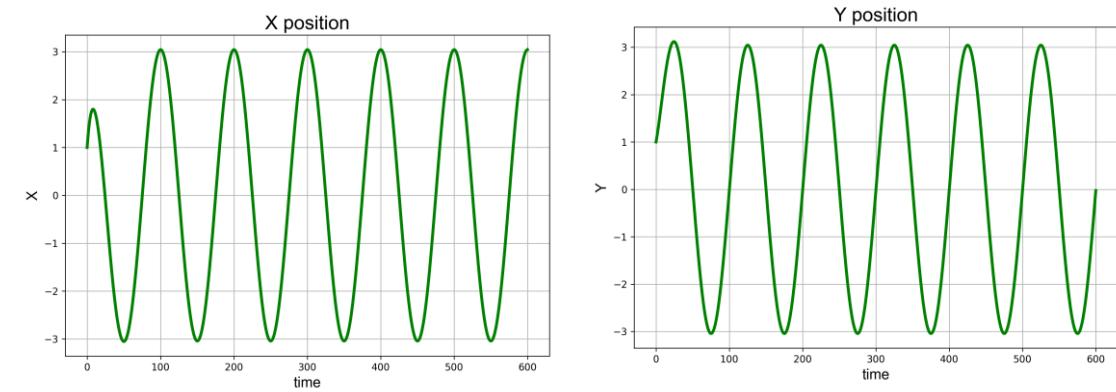


Figura 4.19. Graficas de posición.

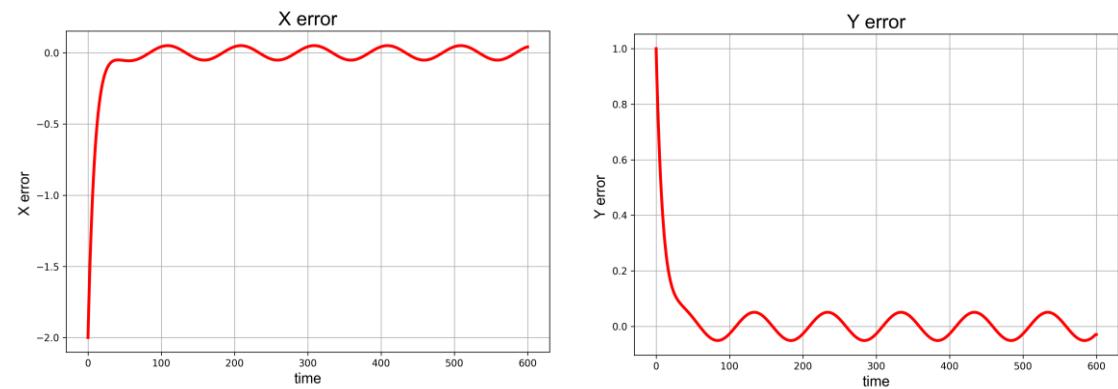
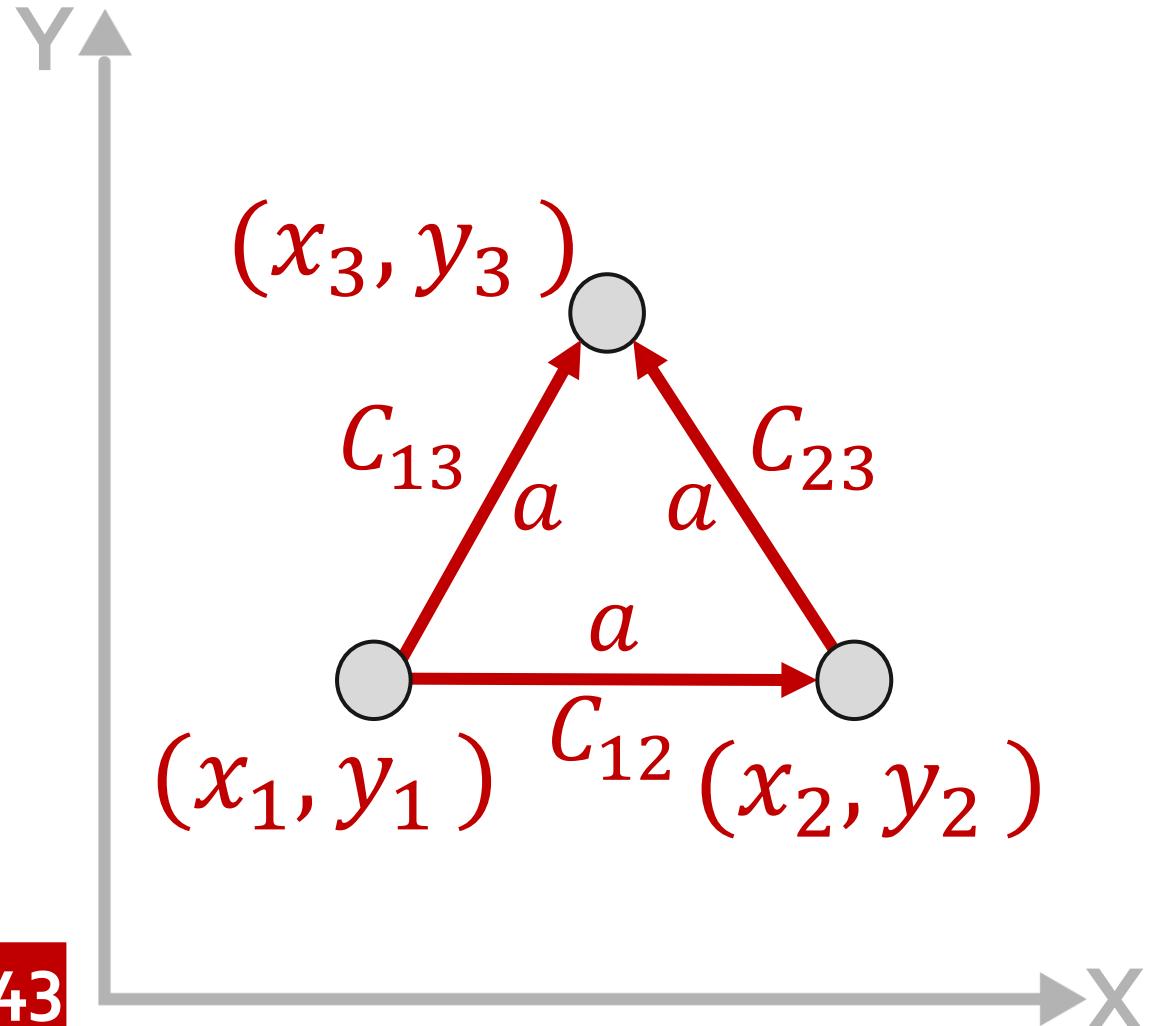


Figura 4.20. Graficas de error.

FORMACIÓN

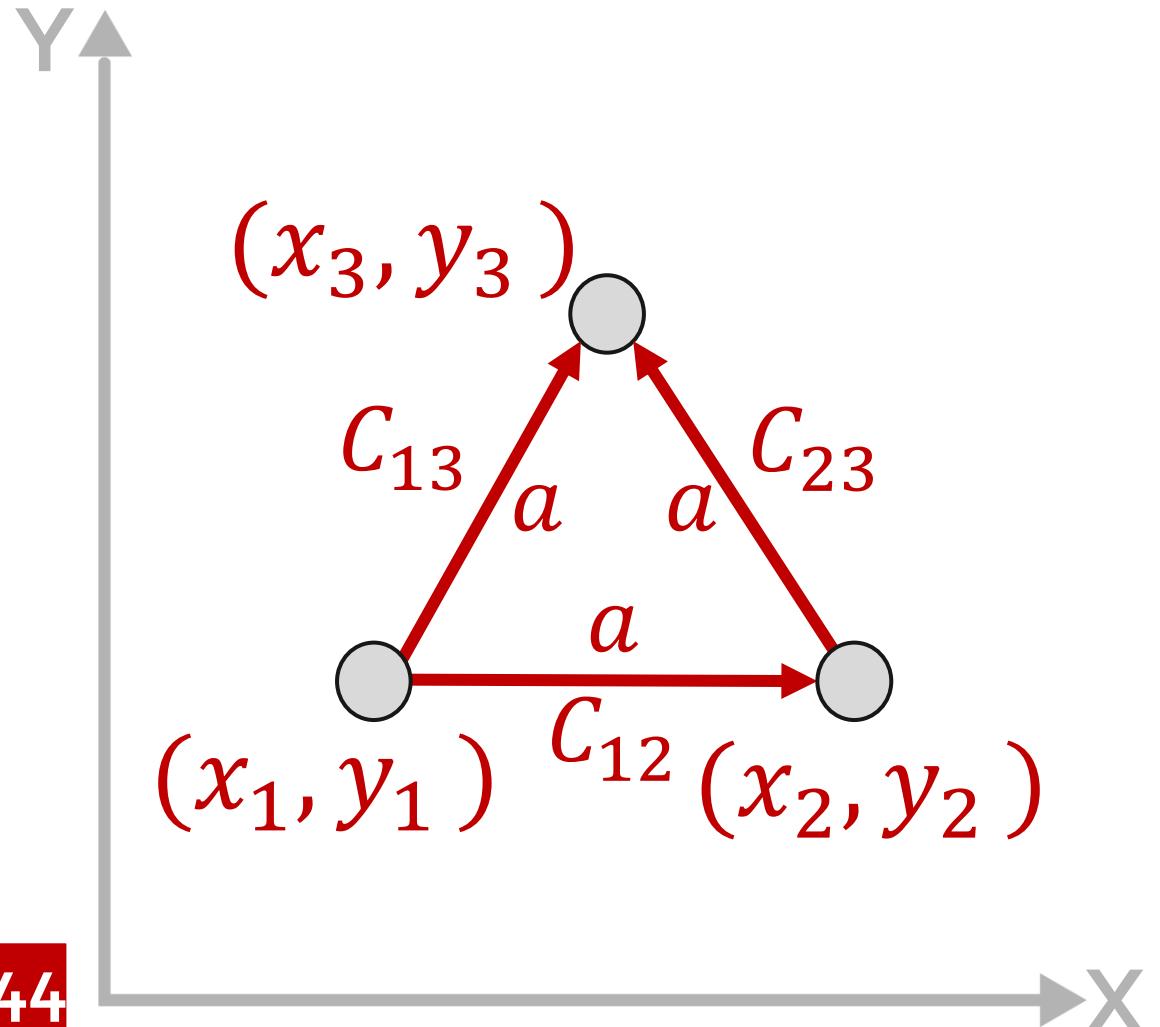


$$C_{12x} = a \quad C_{12y} = 0$$

$$C_{13y} = \sqrt[2]{\frac{3a^2}{4}} \quad C_{13x} = a/2$$

$$C_{23y} = \sqrt[2]{\frac{3a^2}{4}} \quad C_{23x} = -a/2$$

FORMACIÓN



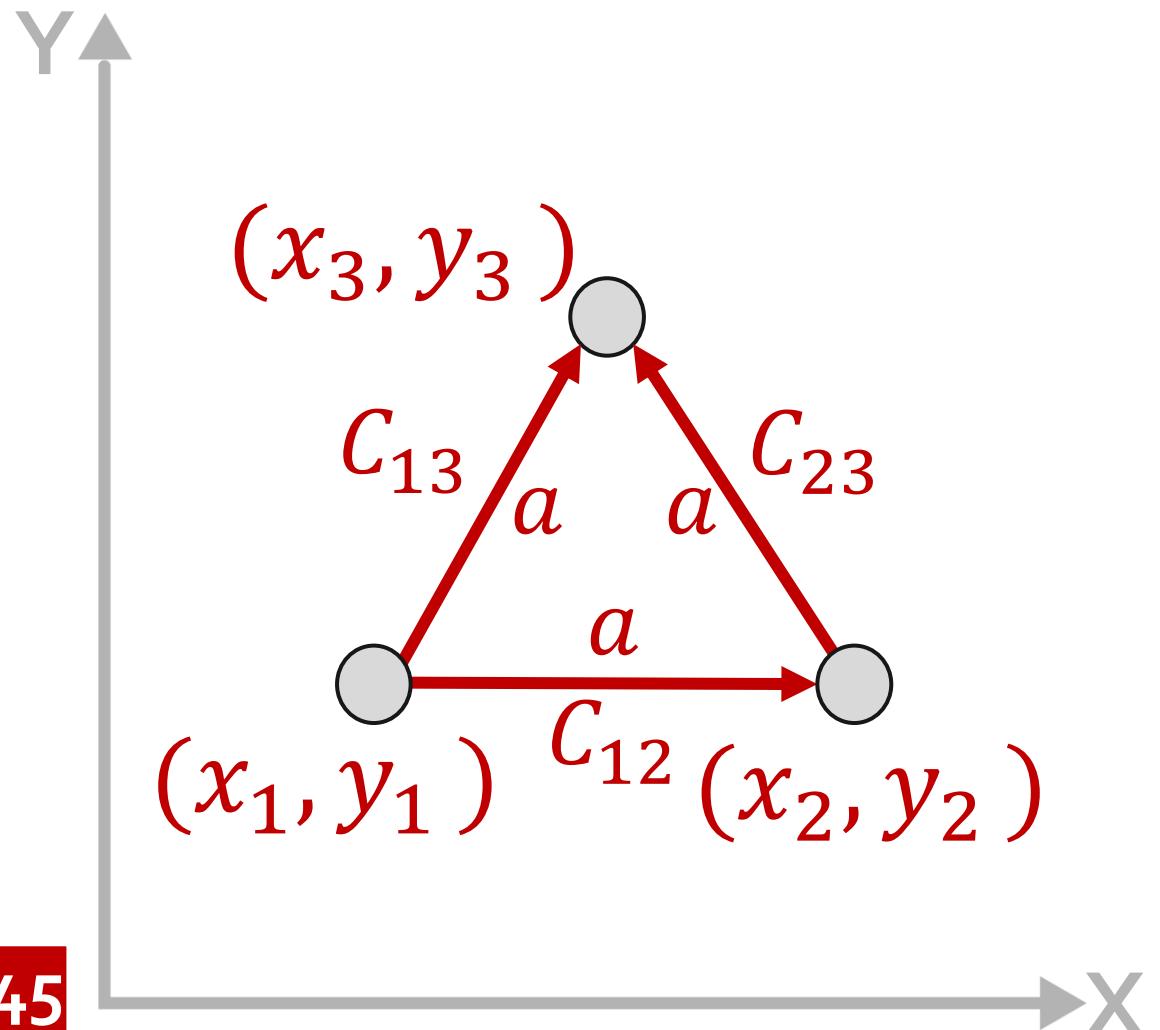
$(x_1, y_1) \rightarrow$ posición actual
 $(x_{1s}, y_{1s}) \rightarrow$ posición deseada
(star point)

$$x_{2s} = x_1 + C_{12}x$$
$$y_{2s} = y_1 + C_{12}y$$

$$x_{3s} = \frac{1}{2}((x_2 + C_{23}x) + (x_1 + C_{13}x))$$
$$y_{3s} = \frac{1}{2}((y_2 + C_{23}y) + (y_1 + C_{13}y))$$

FORMACIÓN

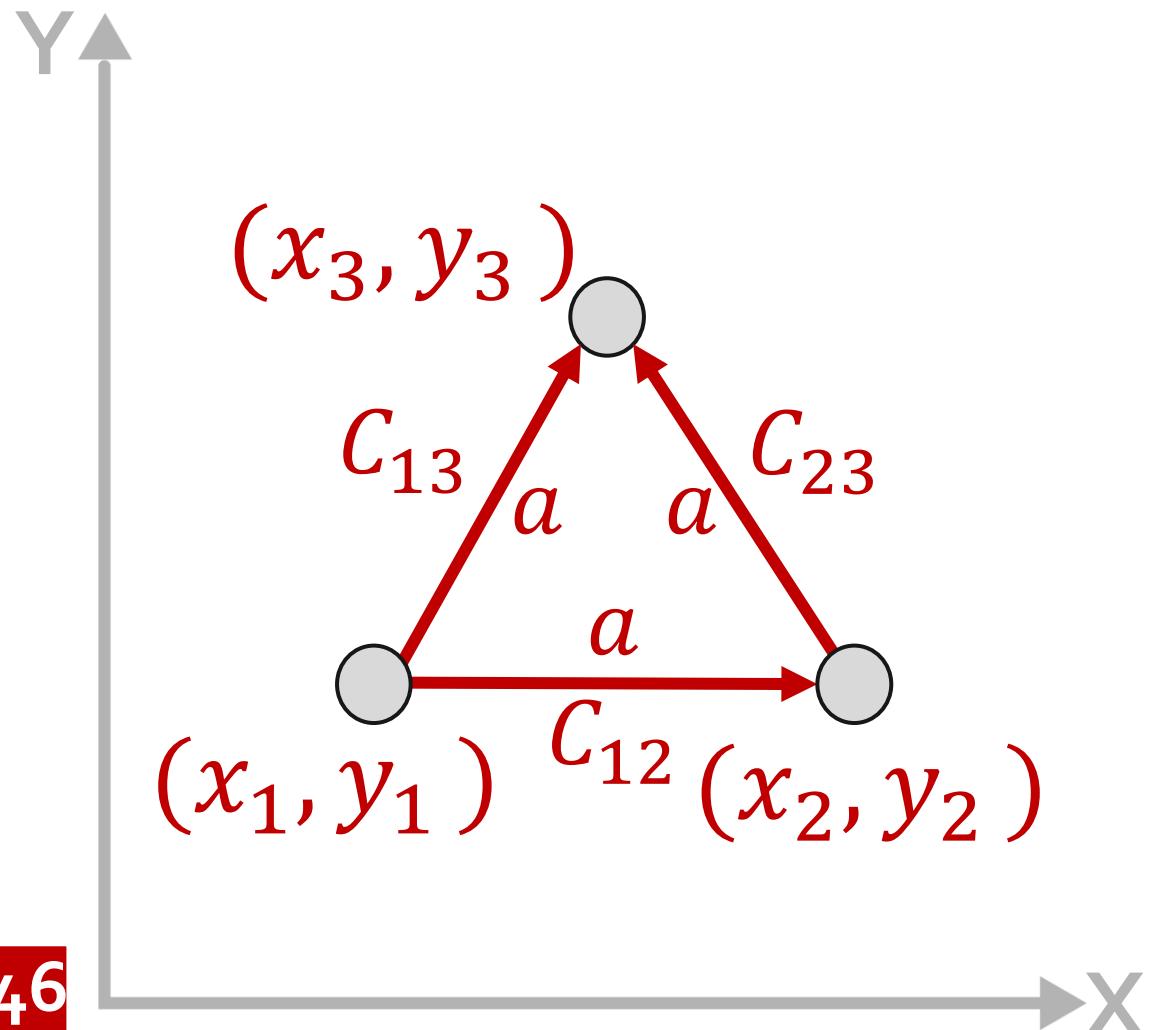
$$k = 1$$



$$\begin{aligned}error_{x_1} &= x_1 - x_{1s} \\error_{y_1} &= y_1 - y_{1s} \\u_{x_1} &= -k * error_{x_1} \\u_{y_1} &= -k * error_{y_1} \\ \\error_{x_2} &= x_2 - x_{2s} \\error_{y_2} &= y_2 - y_{2s} \\u_{x_2} &= -k * error_{x_2} \\u_{y_2} &= -k * error_{y_2}\end{aligned}$$

FORMACIÓN

$$k = 1$$



$$\begin{aligned}error_{x3} &= x_3 - x_{3s} \\error_{y3} &= y_3 - y_{3s} \\u_{x3} &= -k * error_{x3} \\u_{y3} &= -k * error_{y3}\end{aligned}$$

FORMACION

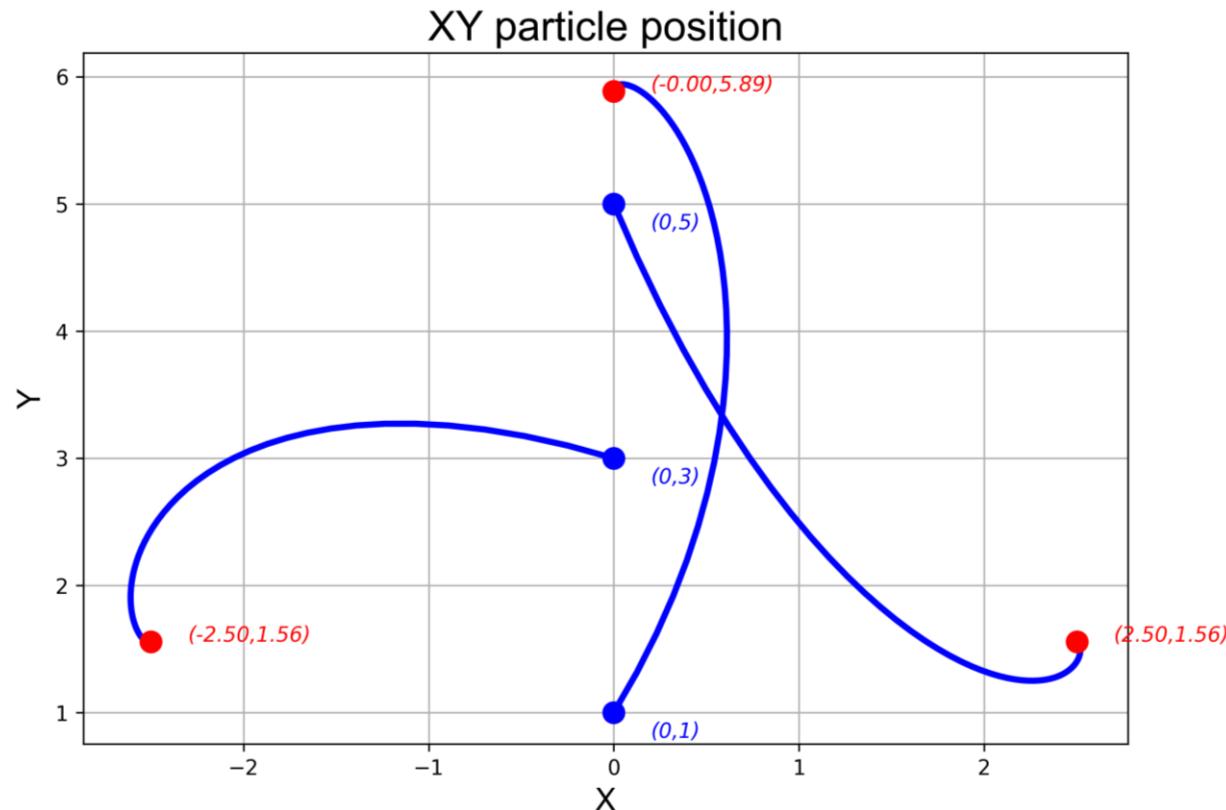


Figura 4.21. Grafica posición XY.

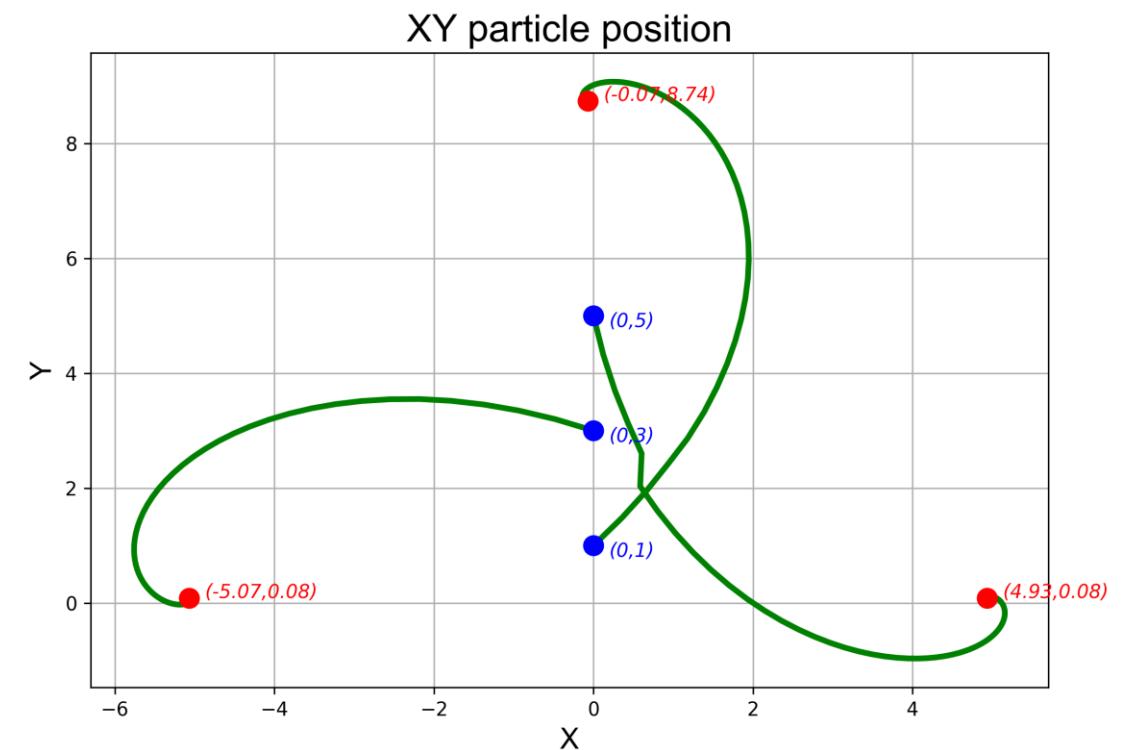


Figura 4.22. Grafica posición XY.

FORMACION

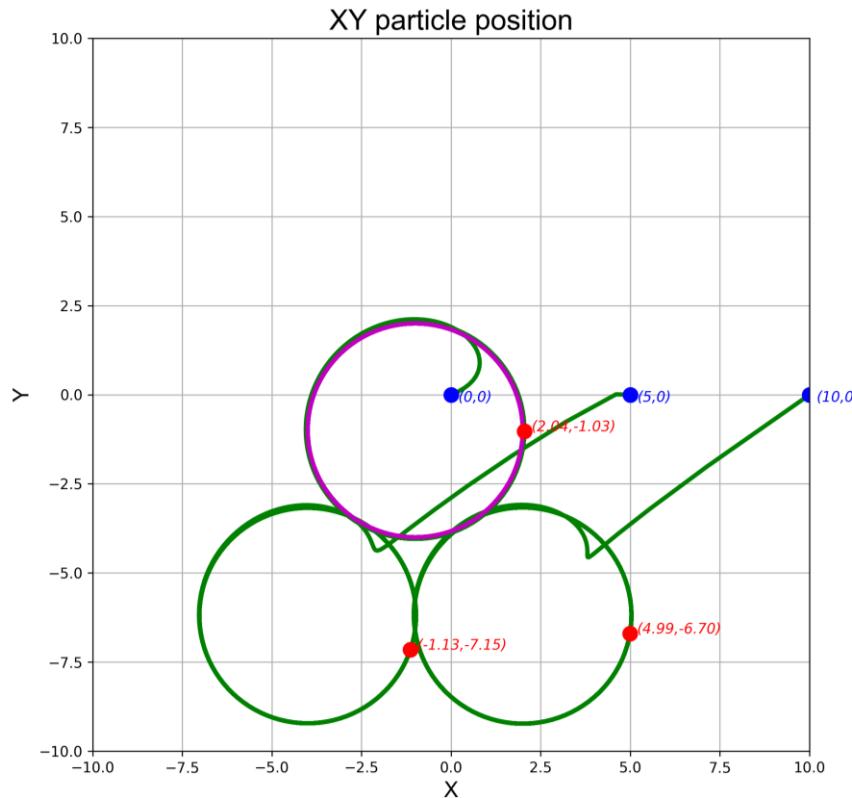


Figura 4.23. Grafica posición XY.

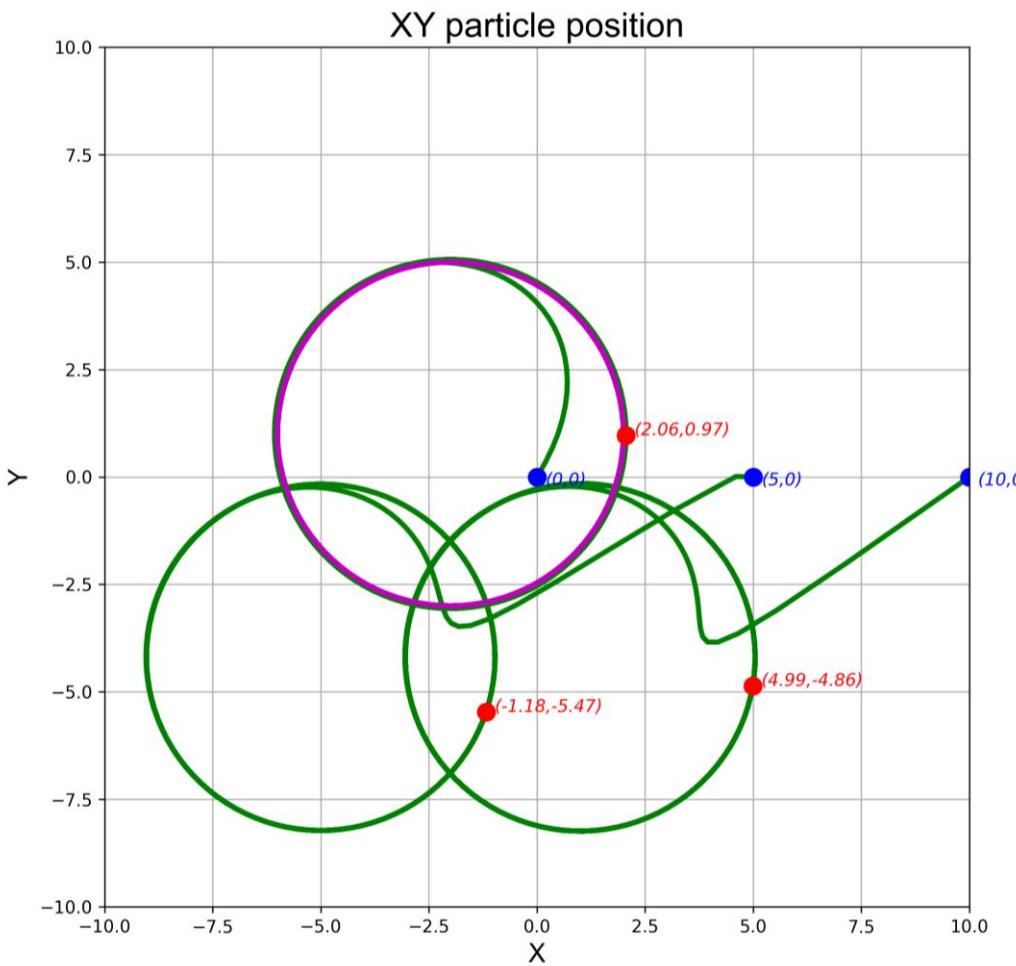
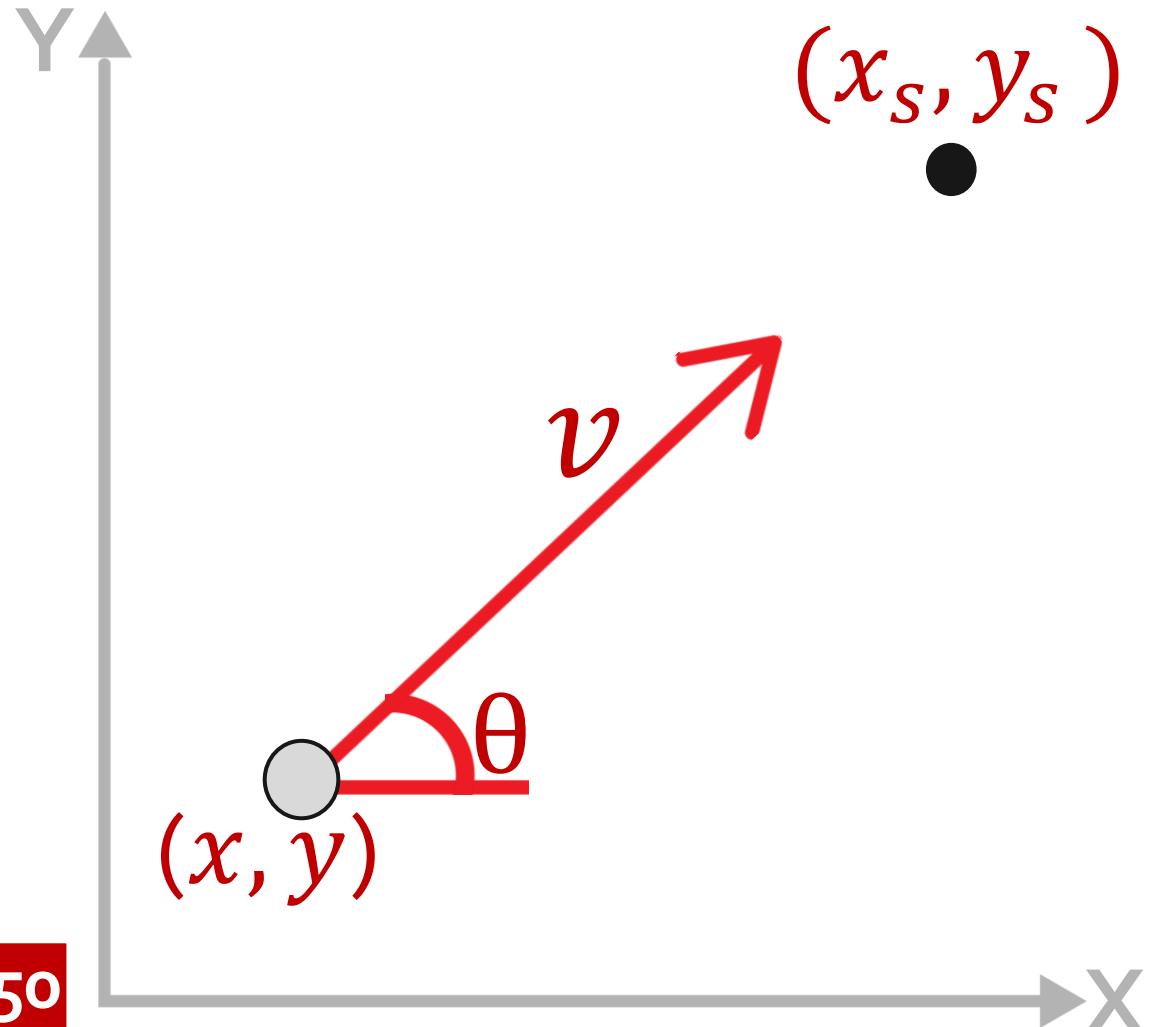


Figura 4.24. Grafica posición XY.

5. MODELO CINEMÁTICO

MODELO PARTÍCULA

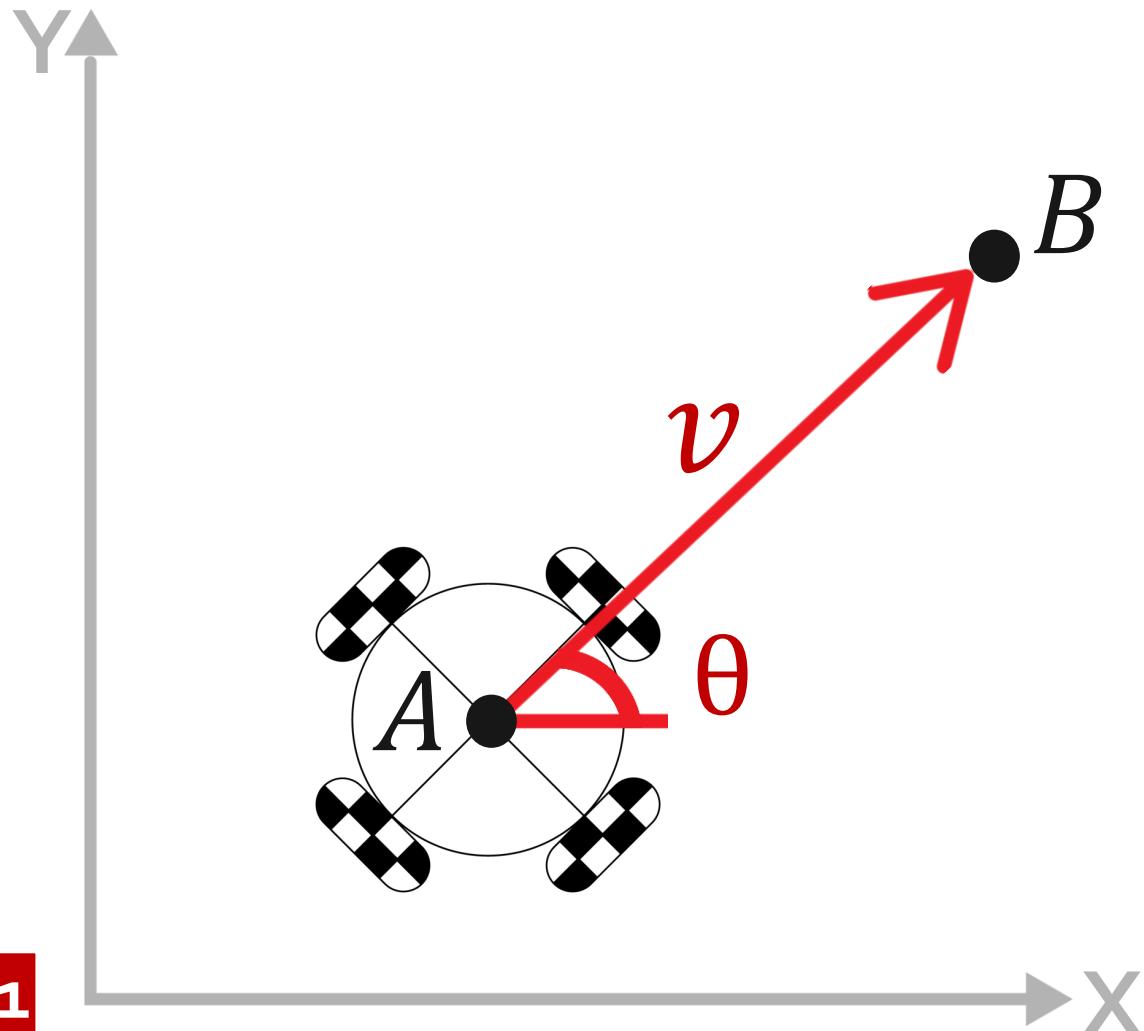


$(x, y) \rightarrow$ posición actual
 $\theta \rightarrow$ orientación

$(x_s, y_s) \rightarrow$ posición deseada
(star point)

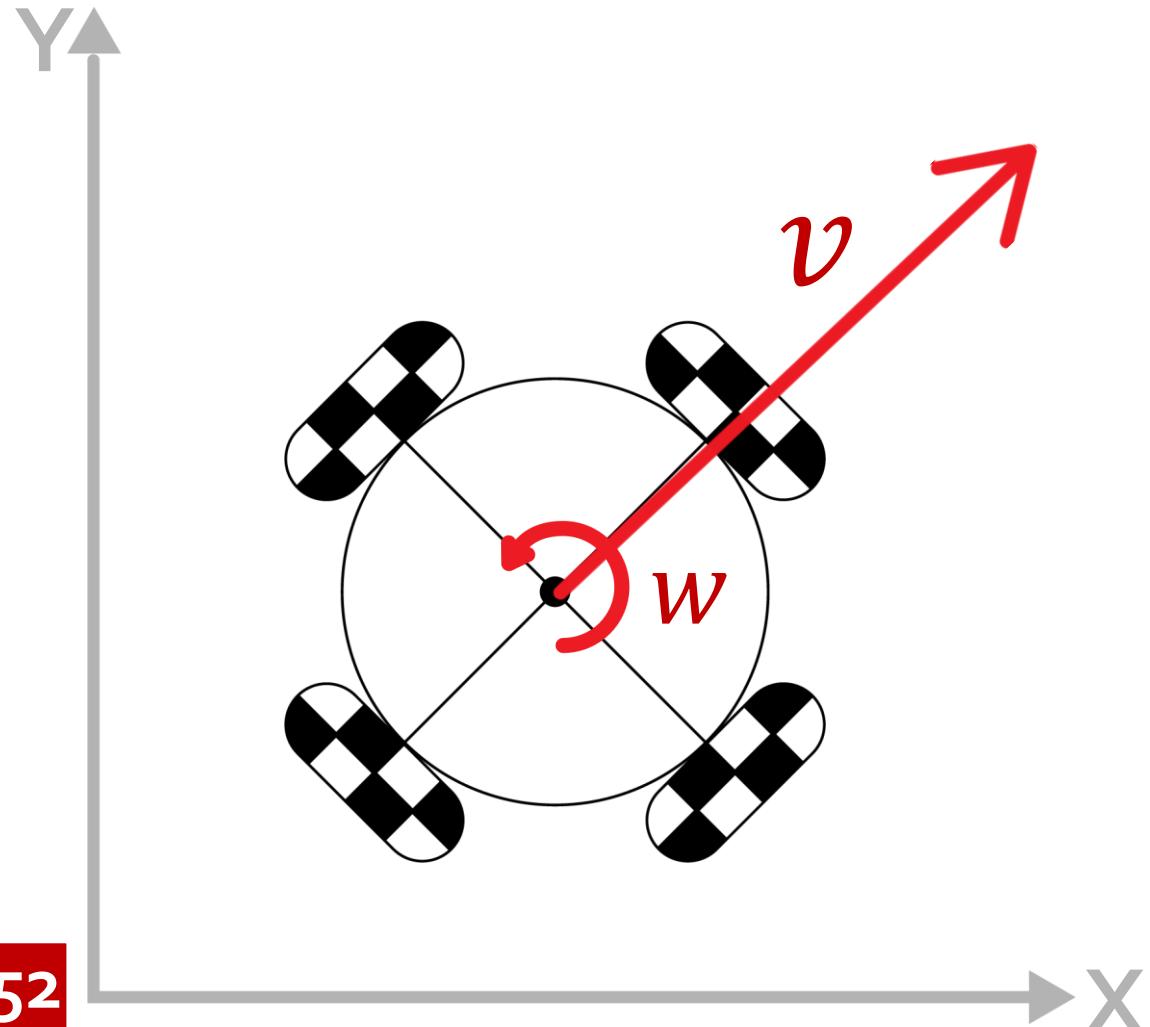
$v \rightarrow$ Velocidad Lineal

MODELO CINEMÁTICO



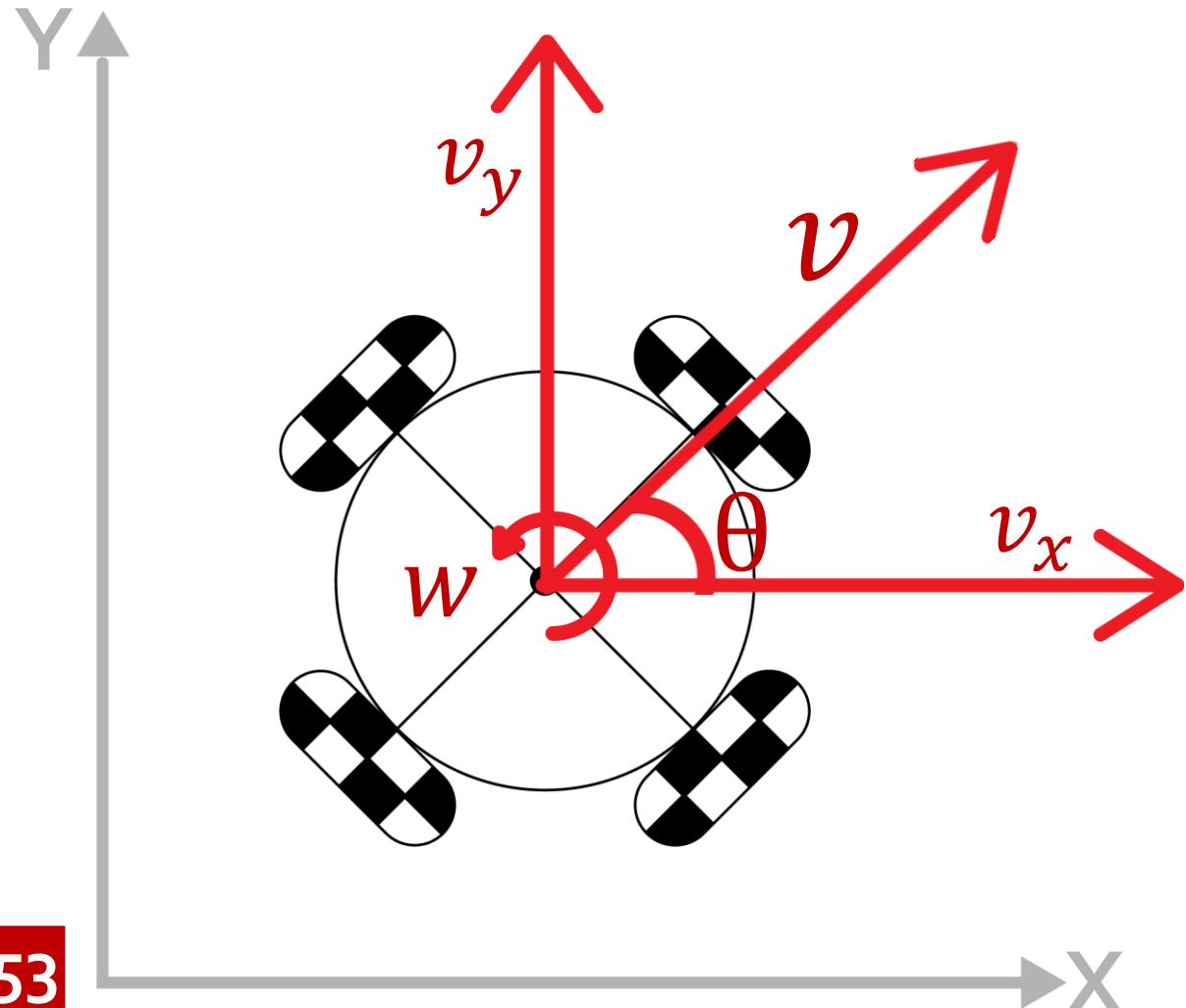
v → Velocidad Lineal
 θ → Orientación

MODELO CINEMÁTICO



v → Velocidad Lineal
 w → Velocidad Angular

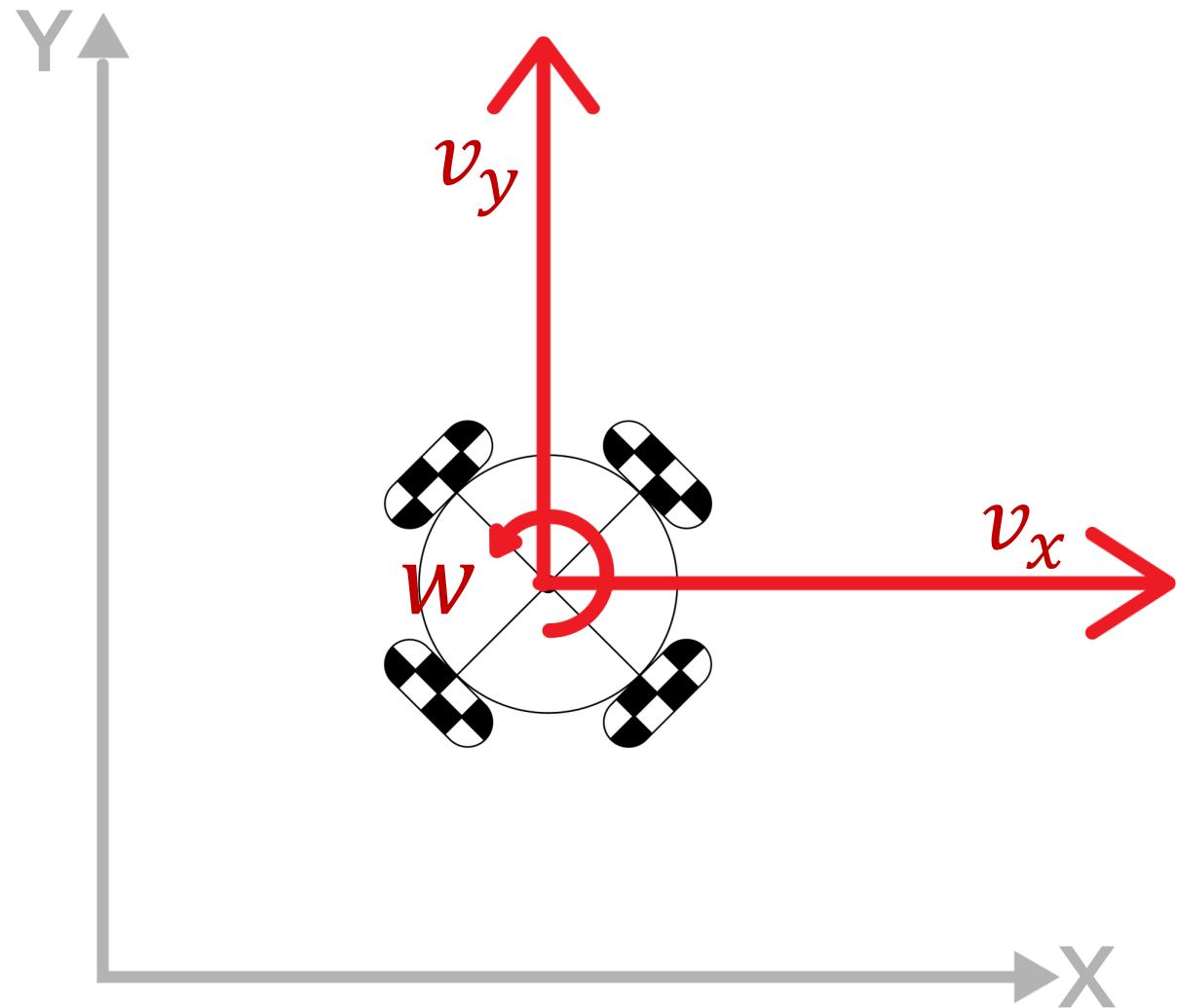
MODELO CINEMATICO



v → Velocidad Lineal
 w → Velocidad Angular
 θ → Orientación

$$v_x = v \cos \theta$$
$$v_y = v \sin \theta$$

MODELO CINEMATICO



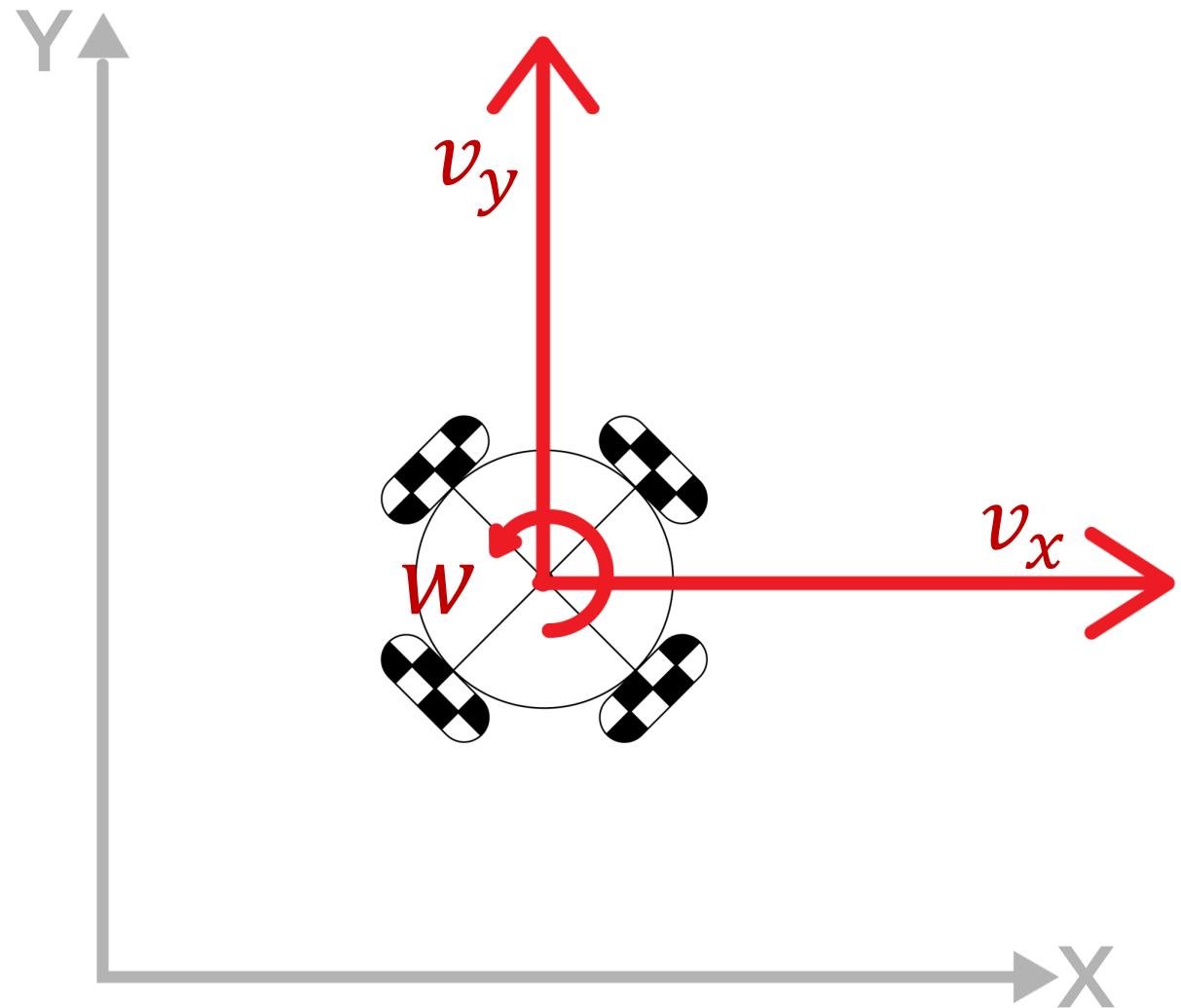
Ejemplo:

$$v_x = v \cos \theta$$

$$v_y = 0$$

$$w = 0$$

MODELO CINEMATICO



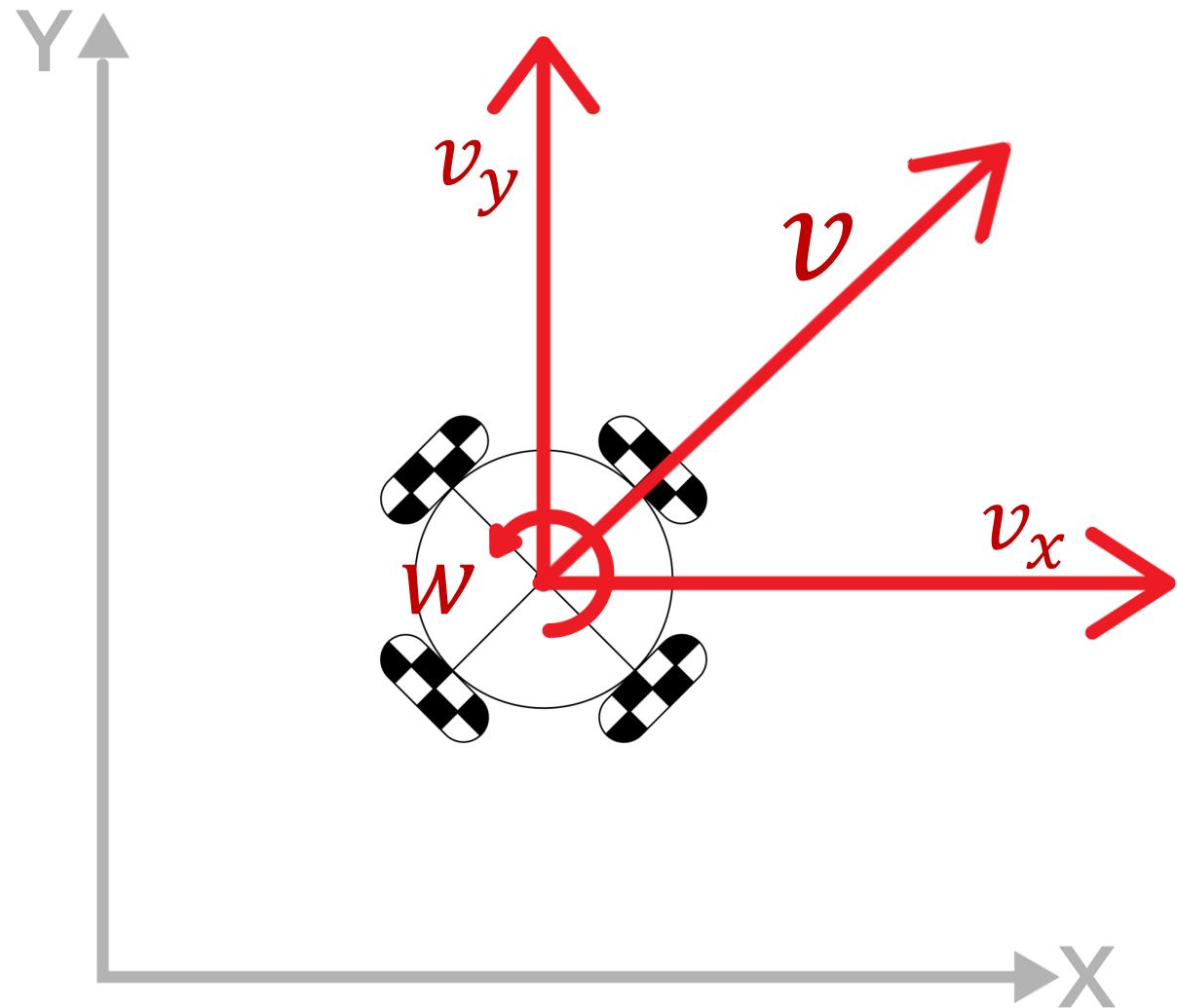
Ejemplo:

$$v_x = 0$$

$$v_y = v \sin \theta$$

$$w = 0$$

MODELO CINEMATICO



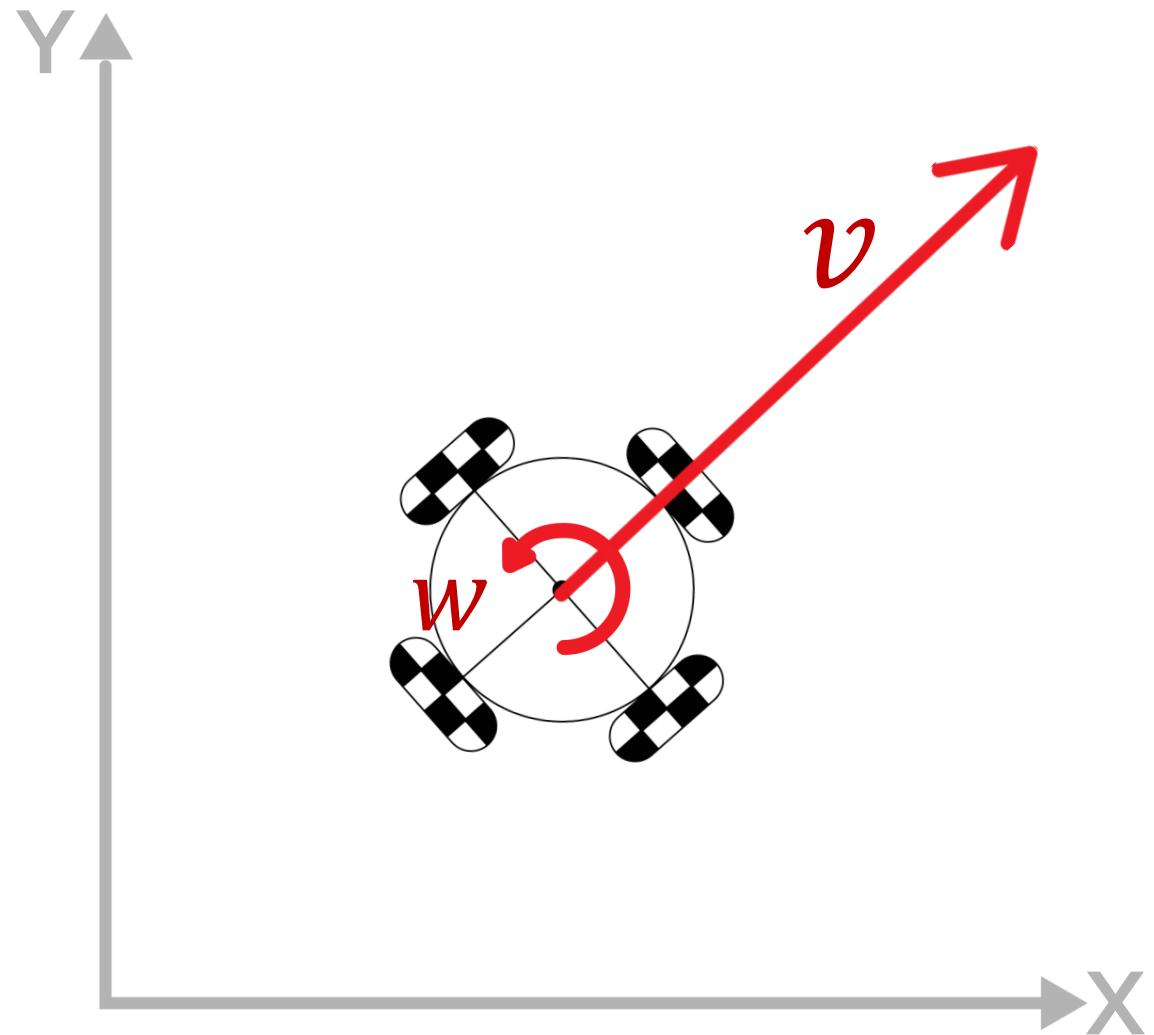
Ejemplo:

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

$$w = 0$$

MODELO CINEMATICO



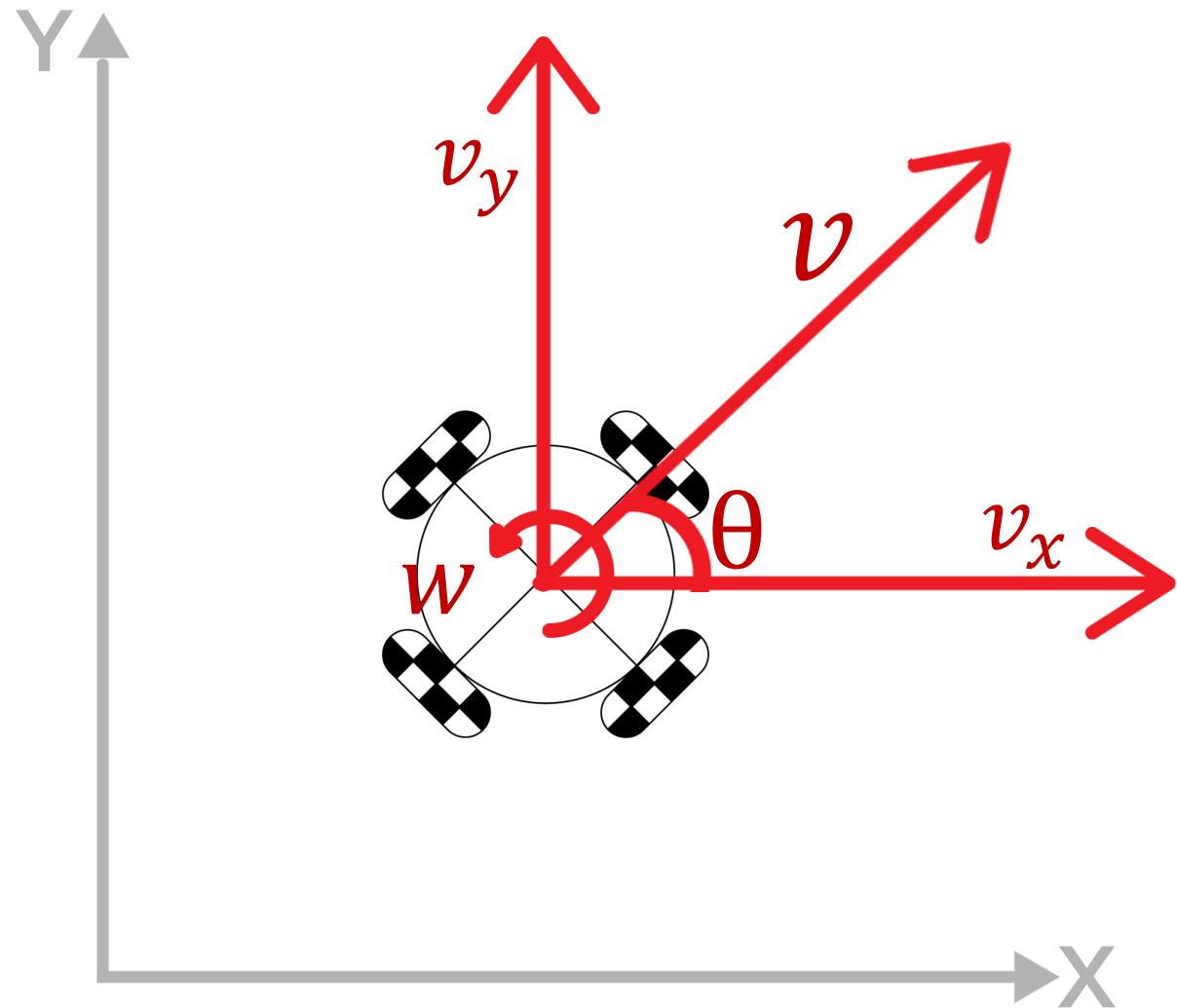
Ejemplo:

$$v_x = 0$$

$$v_y = 0$$

$$w > 0$$

MODELO CINEMATICO



Ejemplo:

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

$$w > 0$$

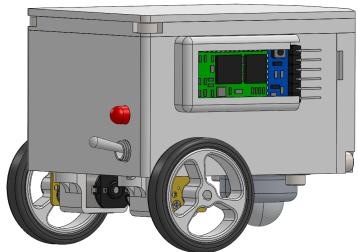


Figura 5.1. Vista isométrica robot uniciclo.

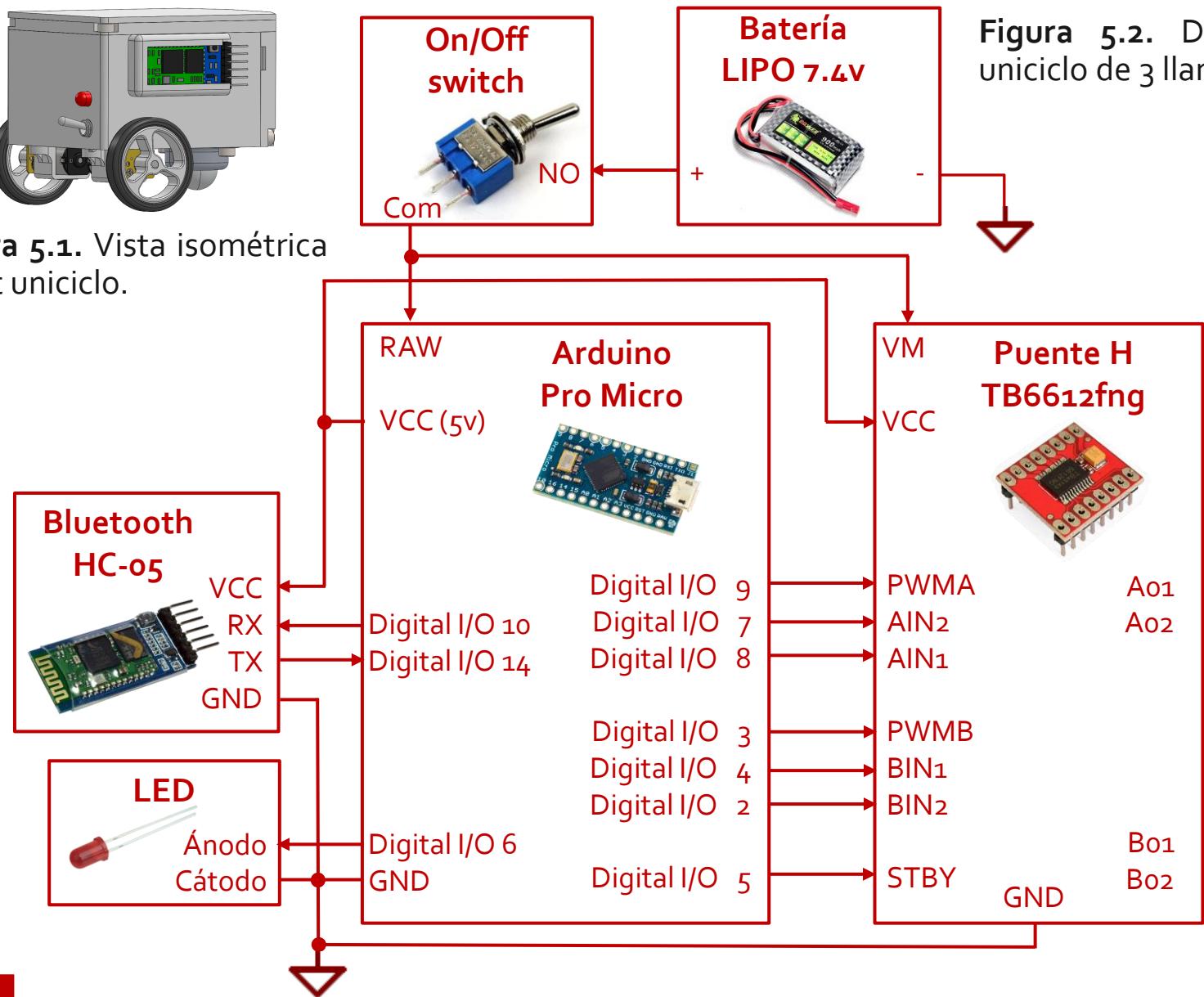


Figura 5.2. Diagrama a bloques de robot uniciclo de 3 llantas: 2 fijas y 1 castor.

$v \rightarrow$ Velocidad Lineal?

$w \rightarrow$ Velocidad Angular?

Pololu micro motorreductor izquierdo

Pololu micro motorreductor derecho

$w_i \rightarrow$ Velocidad angular llanta izquierda

$w_d \rightarrow$ Velocidad angular llanta derecha

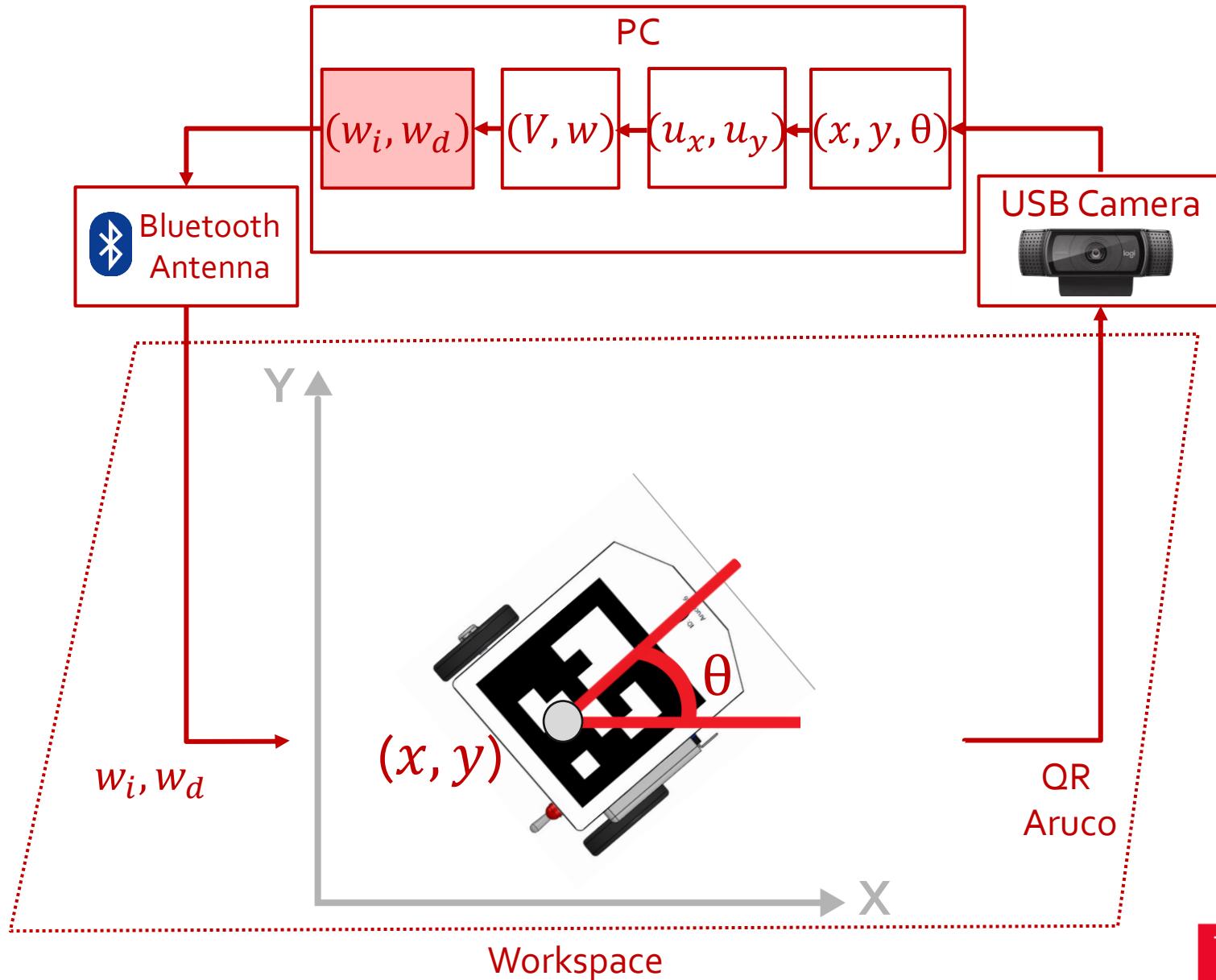
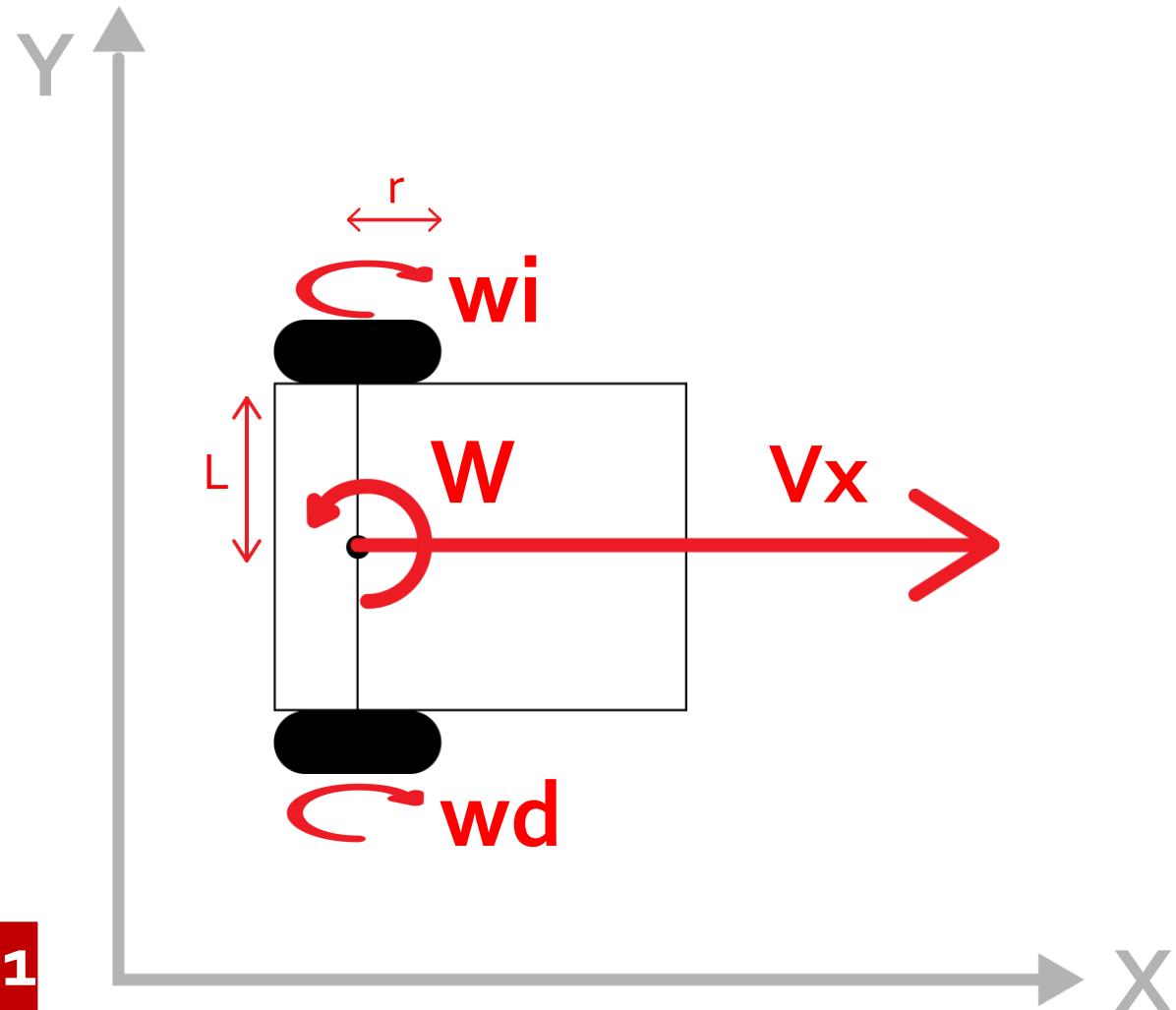


Figura 5.3. Plataforma de experimentación.

WMR 2 LLANTAS: 2 FIJAS

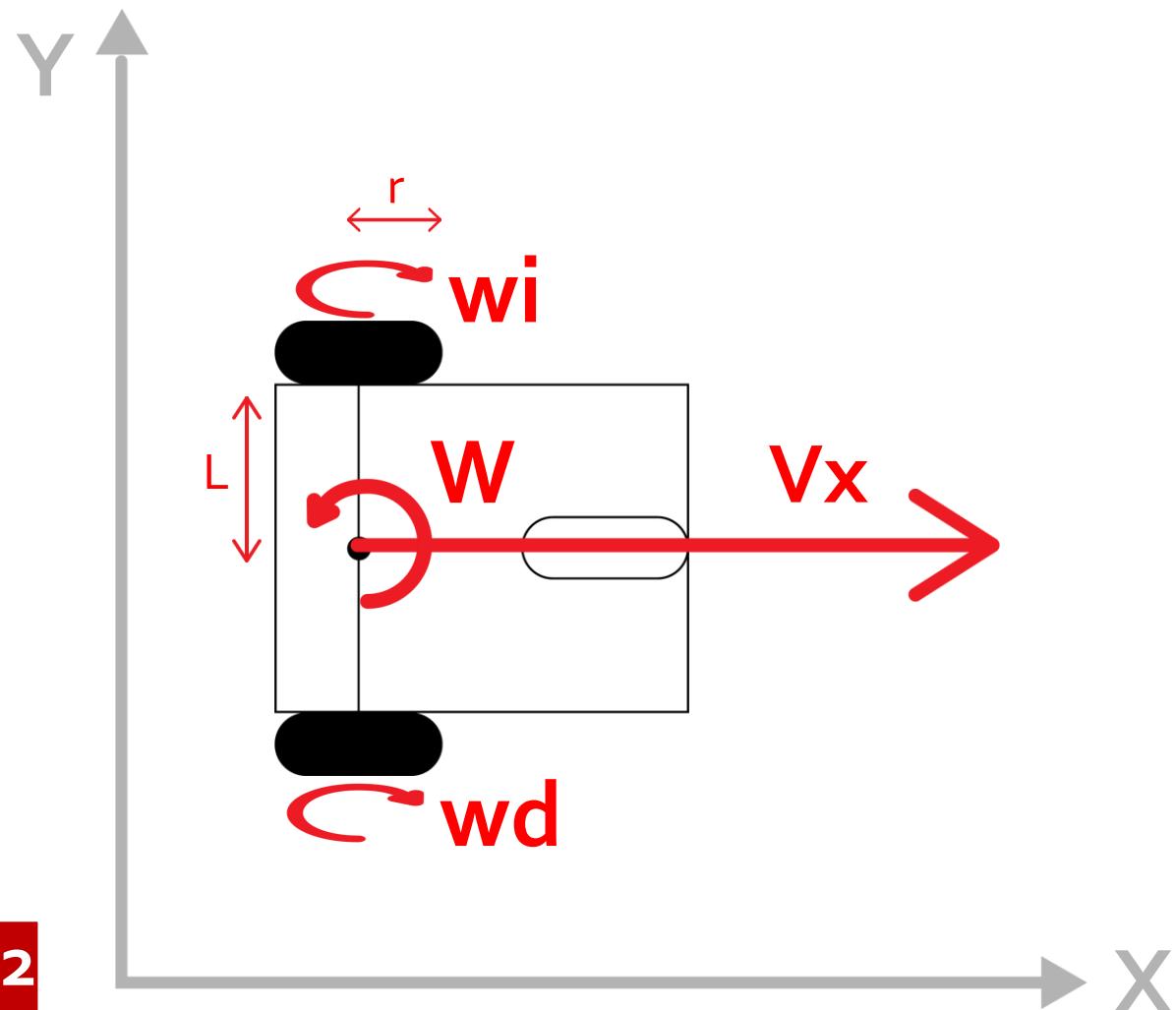


$$\begin{bmatrix} w_i \\ w_d \end{bmatrix} = \left(\frac{1}{r} \right) \begin{bmatrix} 1 & -L \\ 1 & L \end{bmatrix} \begin{bmatrix} v_x \\ w \end{bmatrix}$$

$$w_i = \frac{1}{r} (v_x - Lw)$$

$$w_d = \frac{1}{r} (v_x + Lw)$$

WMR 3 LLANTAS: 2 FIJAS Y 1 CASTOR

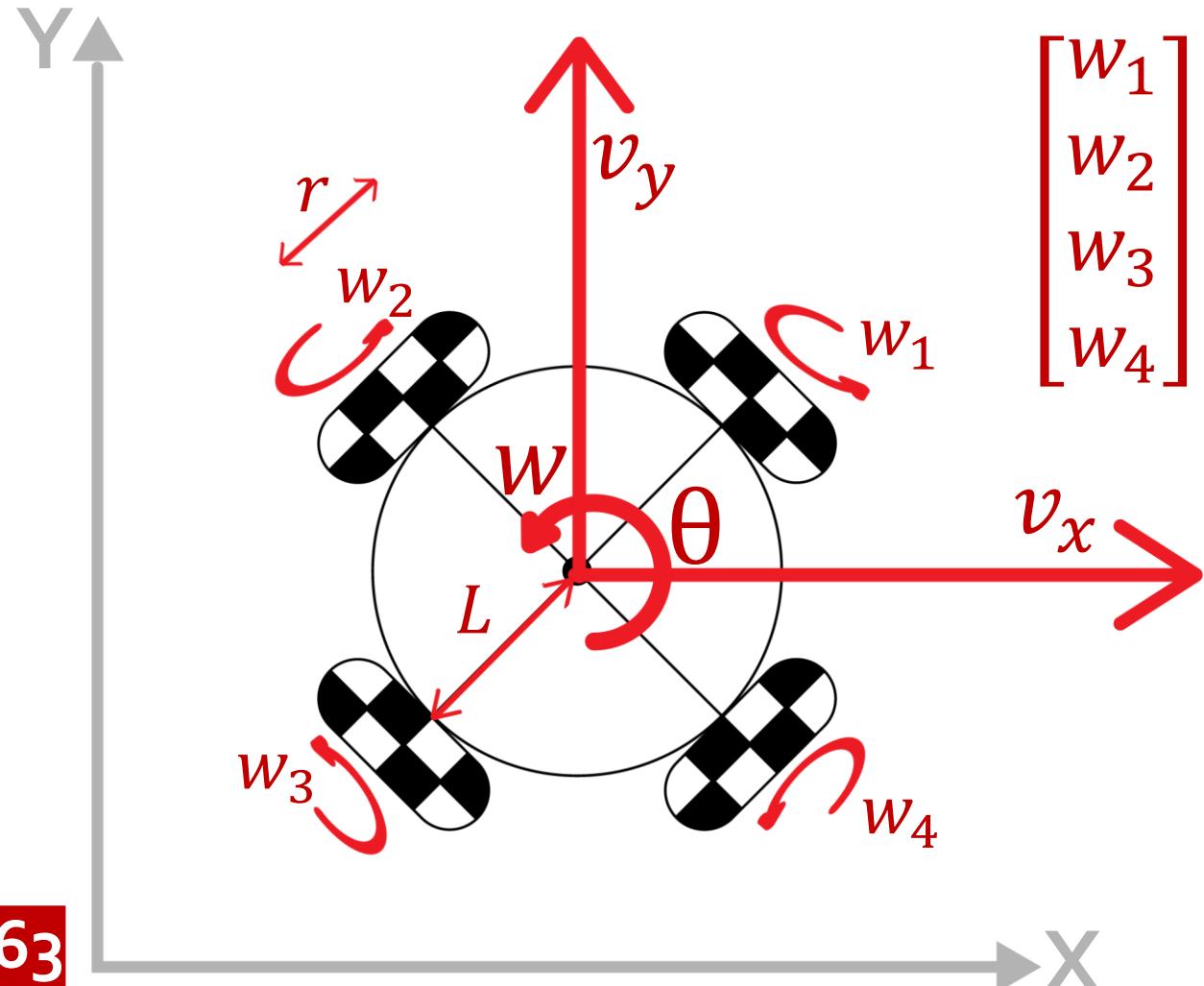


$$\begin{bmatrix} w_i \\ w_d \end{bmatrix} = \left(\frac{1}{r} \right) \begin{bmatrix} 1 & -L \\ 1 & L \end{bmatrix} \begin{bmatrix} v_x \\ w \end{bmatrix}$$

$$w_i = \frac{1}{r} (v_x - Lw)$$

$$w_d = \frac{1}{r} (v_x + Lw)$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES 45°

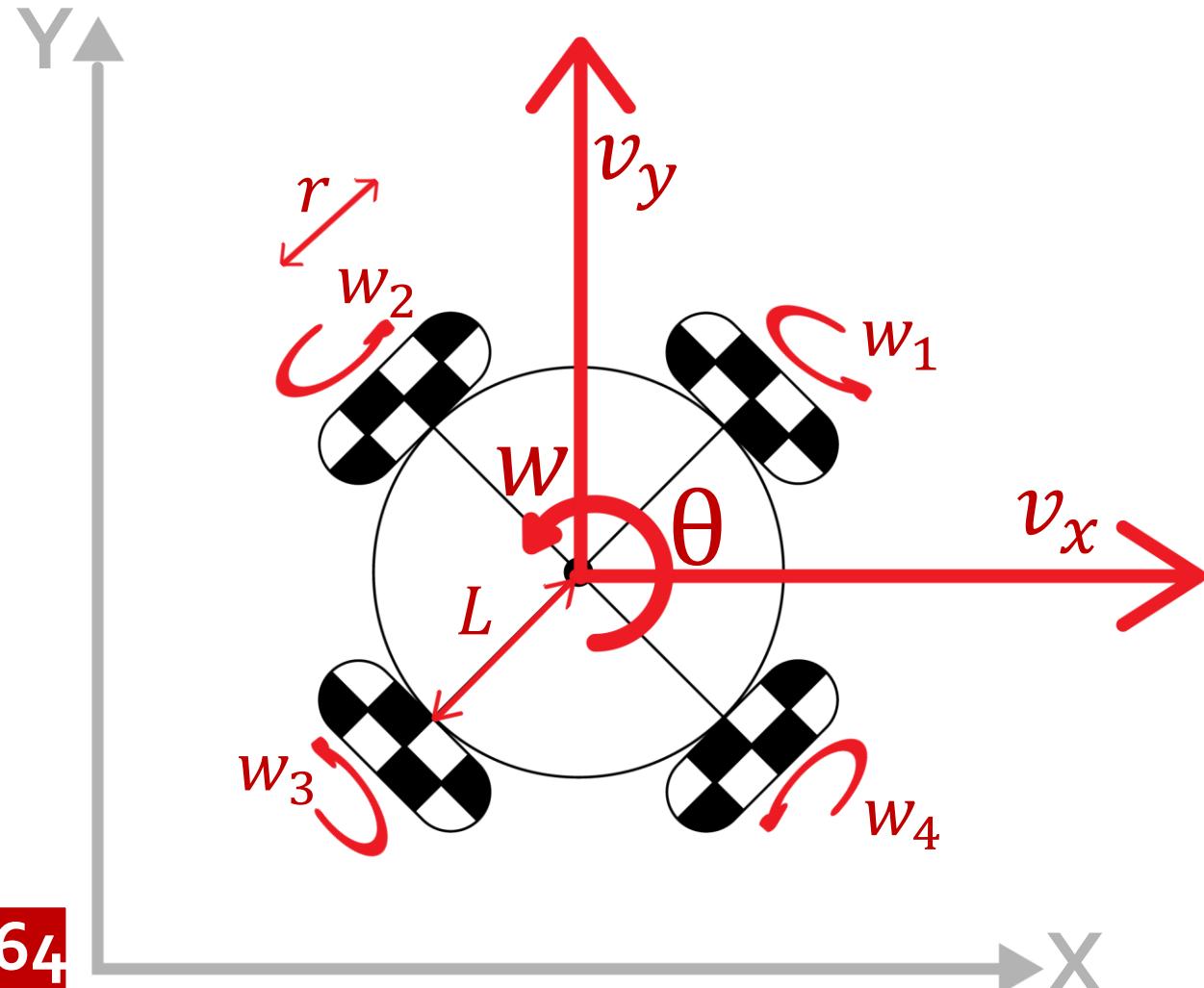


$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \left(\frac{1}{r}\right) \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -L \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -L \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -L \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & -L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}$$

$$v_x = v \cos \theta$$

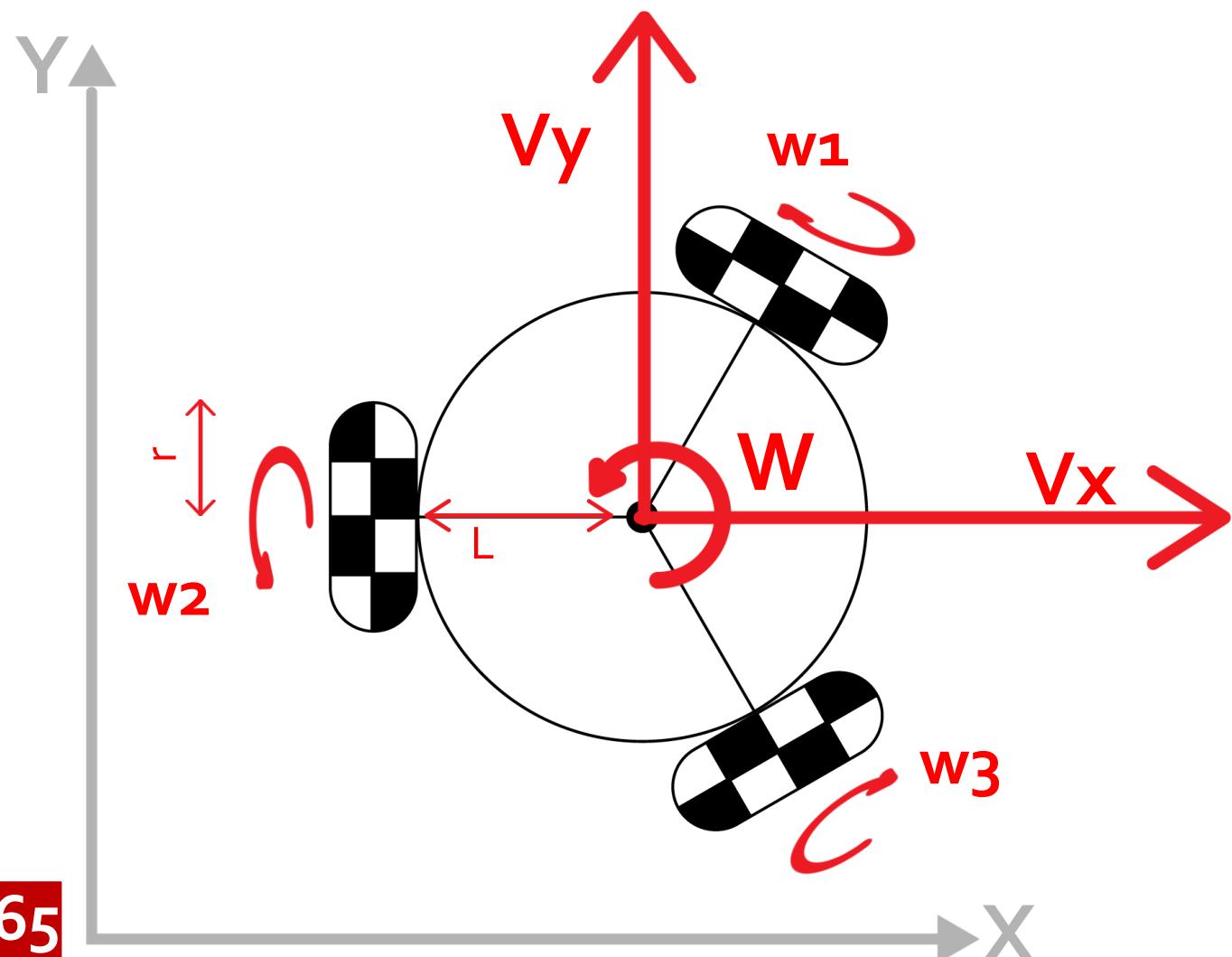
$$v_y = v \sin \theta$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES 45°



$$w_1 = \frac{1}{r} (v_x \sqrt[2]{2} - v_y \sqrt[2]{2} - Lw)$$
$$w_2 = \frac{1}{r} (v_x \sqrt[2]{2} + v_y \sqrt[2]{2} - Lw)$$
$$w_3 = \frac{1}{r} (-v_x \sqrt[2]{2} + v_y \sqrt[2]{2} - Lw)$$
$$w_4 = \frac{1}{r} (-v_x \sqrt[2]{2} - v_y \sqrt[2]{2} - Lw)$$

WMR 3 LLANTAS: 3 OMNIDIRECCIONALES

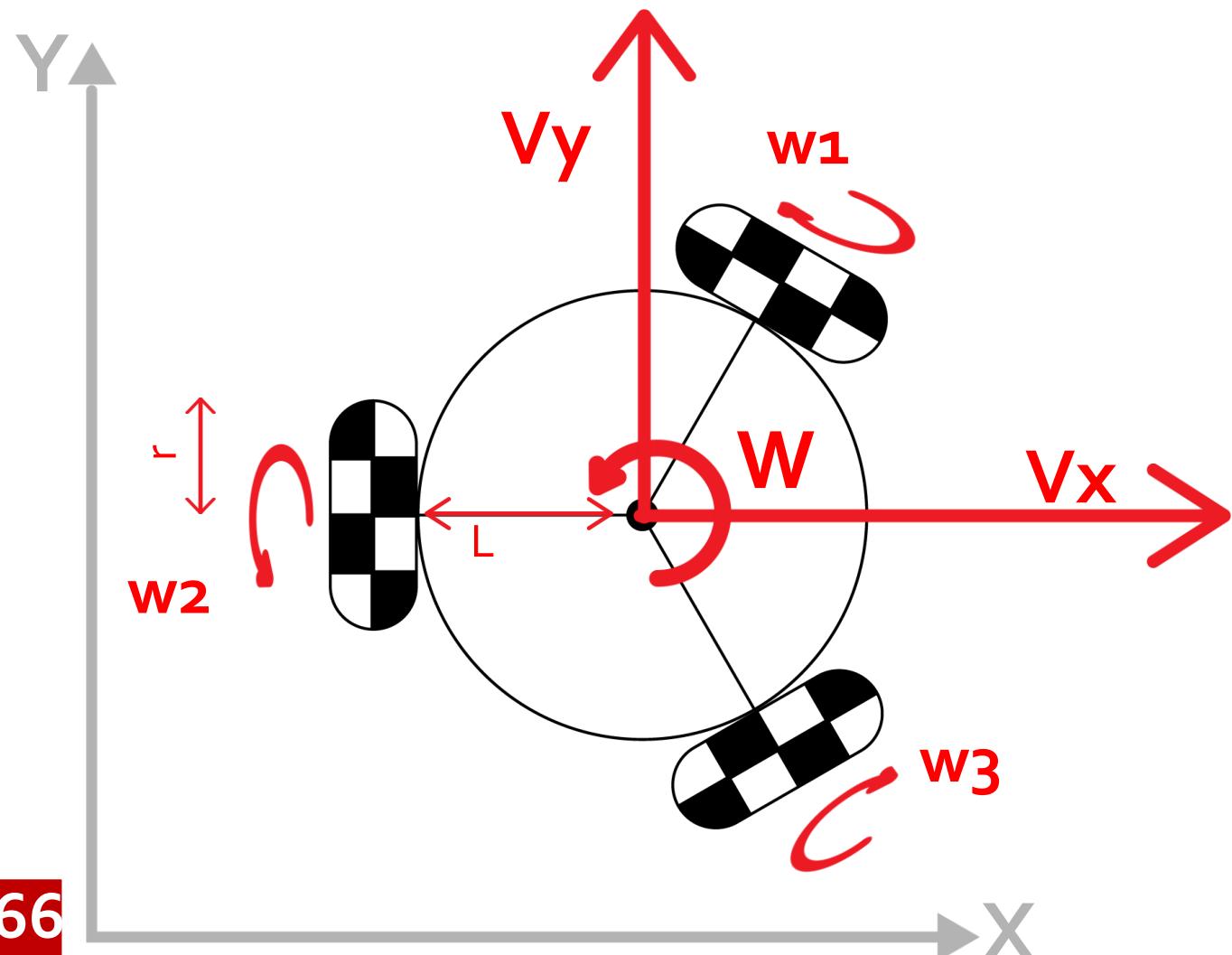


$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \left(\frac{1}{r}\right) \begin{bmatrix} -\frac{2\sqrt{3}}{2} & \frac{1}{2} & L \\ 0 & -1 & L \\ \frac{2\sqrt{3}}{2} & \frac{1}{2} & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}$$

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

WMR 3 LLANTAS: 3 OMNIDIRECCIONALES

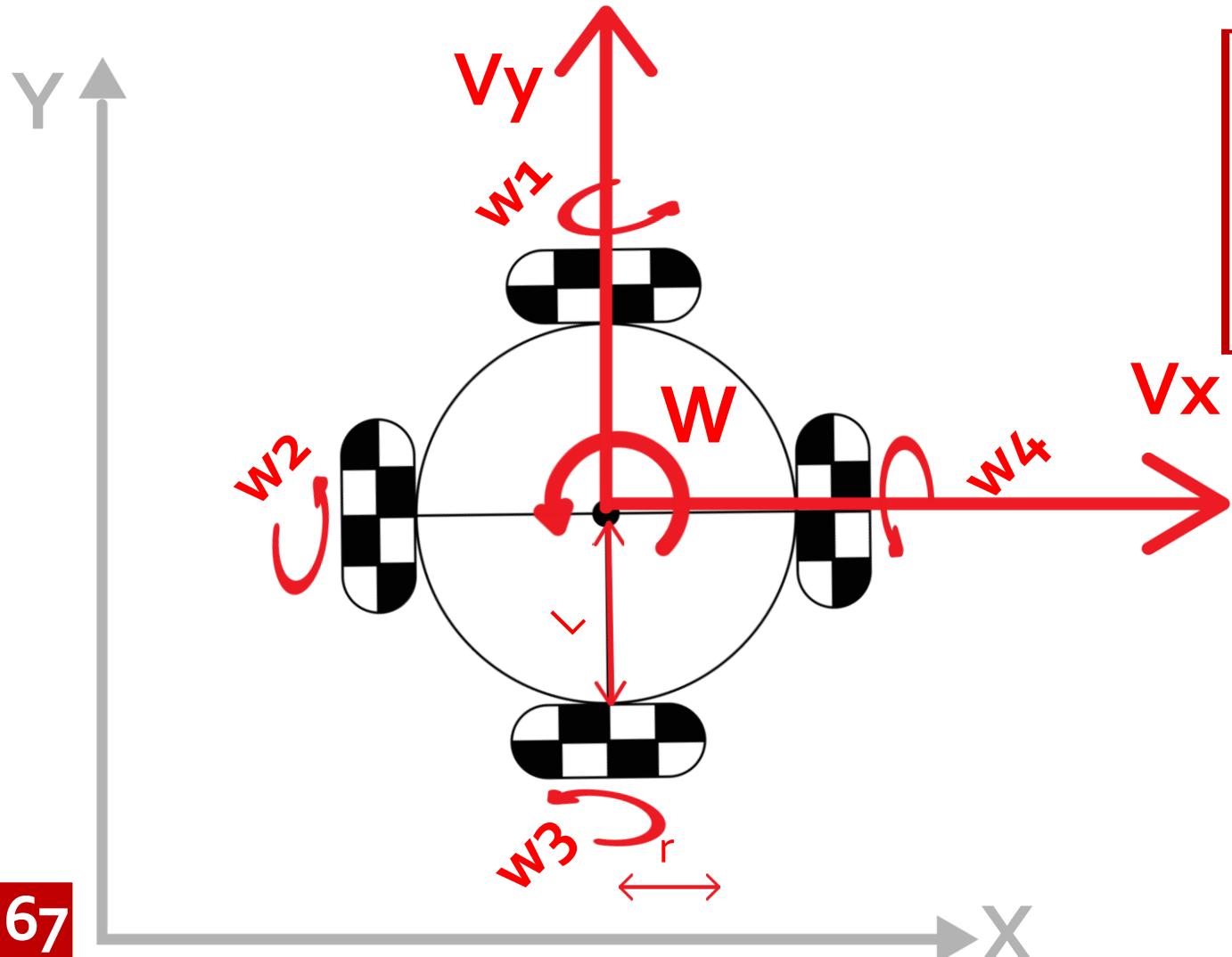


$$w_1 = \frac{1}{r} \left(\frac{-v_x^2 \sqrt{3}}{2} + \frac{v_y}{2} + Lw \right)$$

$$w_2 = \frac{1}{r} (-v_y + Lw)$$

$$w_3 = \frac{1}{r} \left(\frac{v_x^2 \sqrt{3}}{2} + \frac{v_y}{2} + Lw \right)$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES 0°

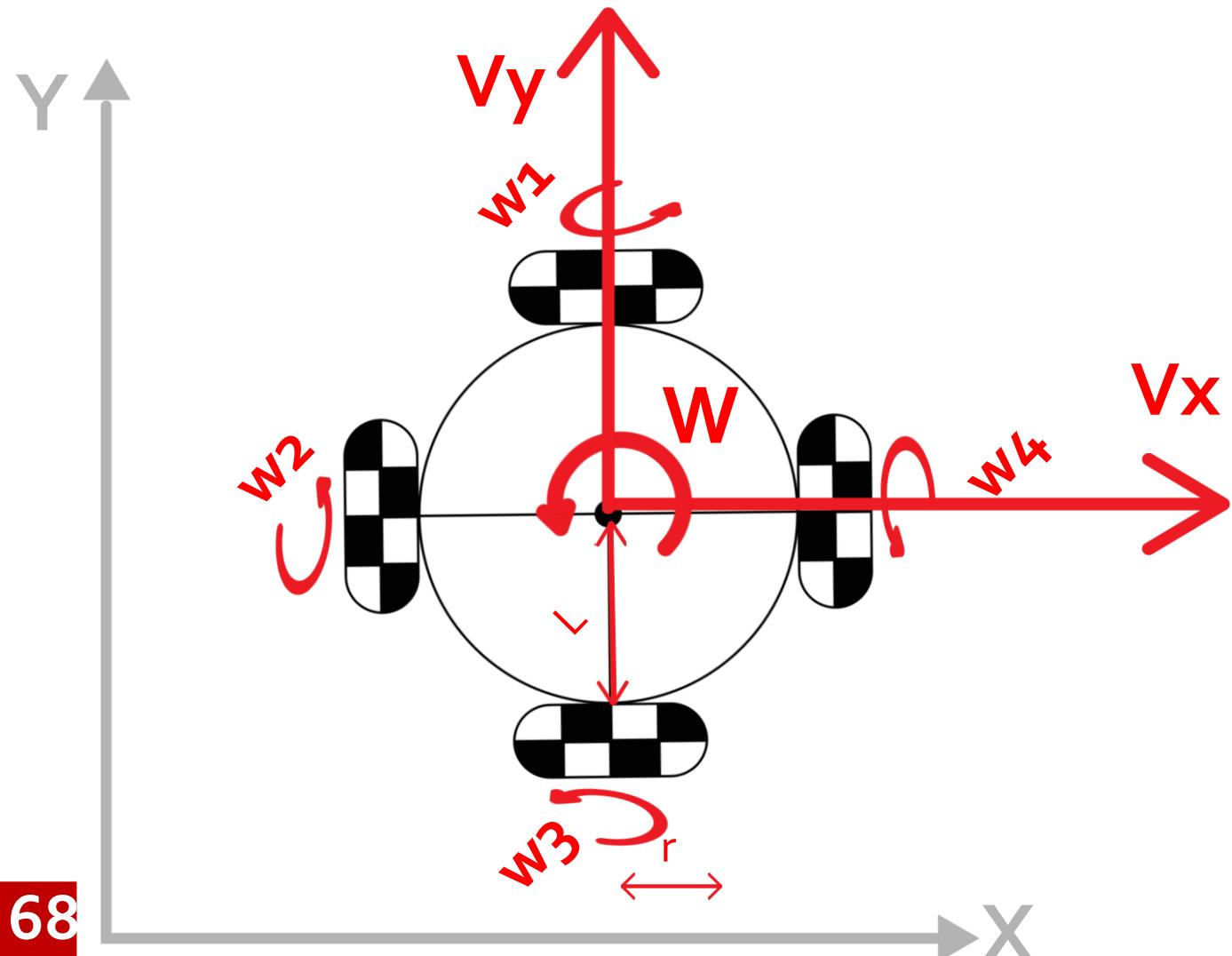


$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \left(\frac{1}{r}\right) \begin{bmatrix} 1 & 0 & -L \\ 0 & 1 & -L \\ -1 & 0 & -L \\ 0 & -1 & -L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}$$

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES 0°



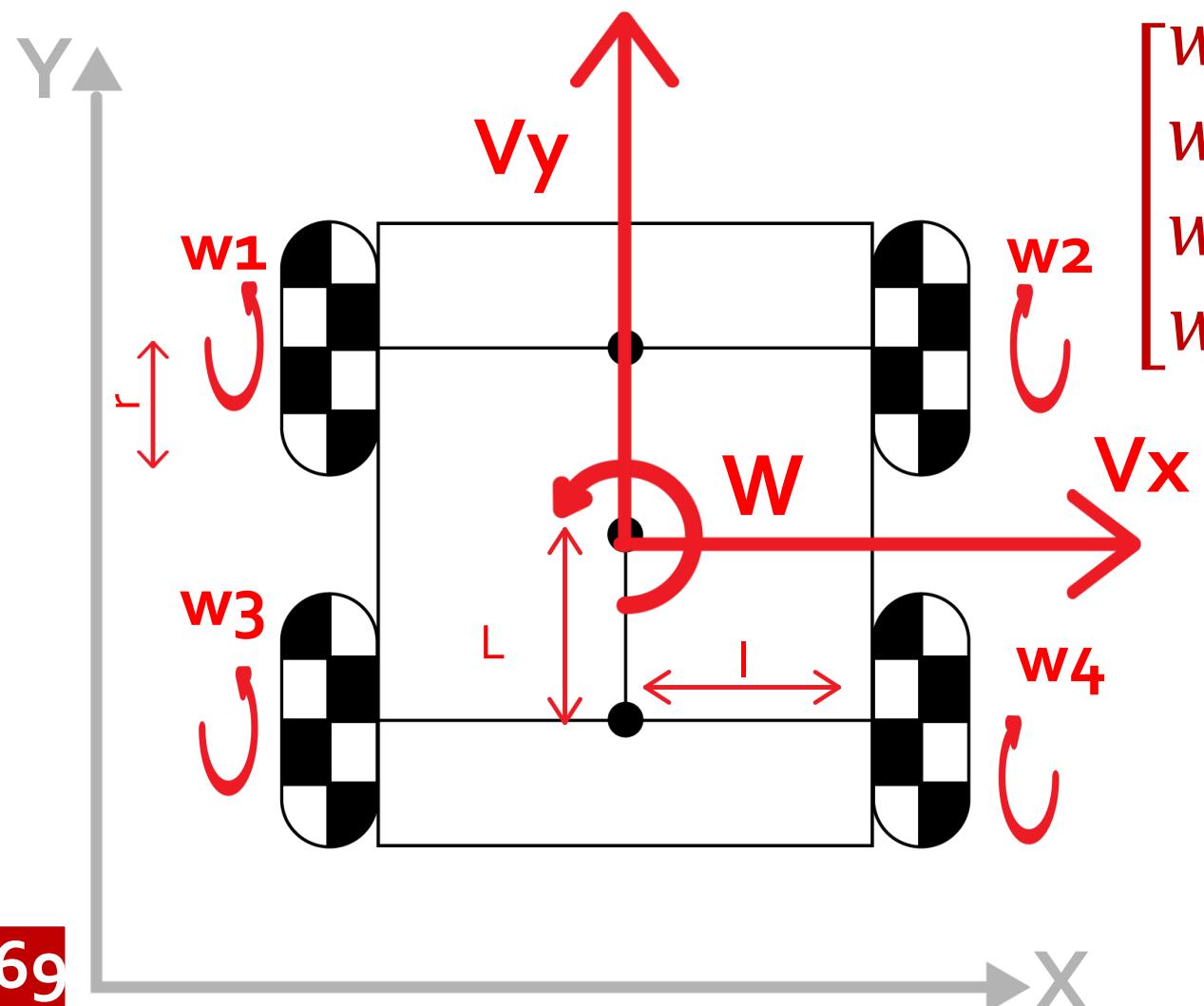
$$w_1 = \frac{1}{r} (v_x - Lw)$$

$$w_2 = \frac{1}{r} (v_y - Lw)$$

$$w_3 = \frac{1}{r} (-v_x - Lw)$$

$$w_4 = \frac{1}{r} (-v_y - Lw)$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES

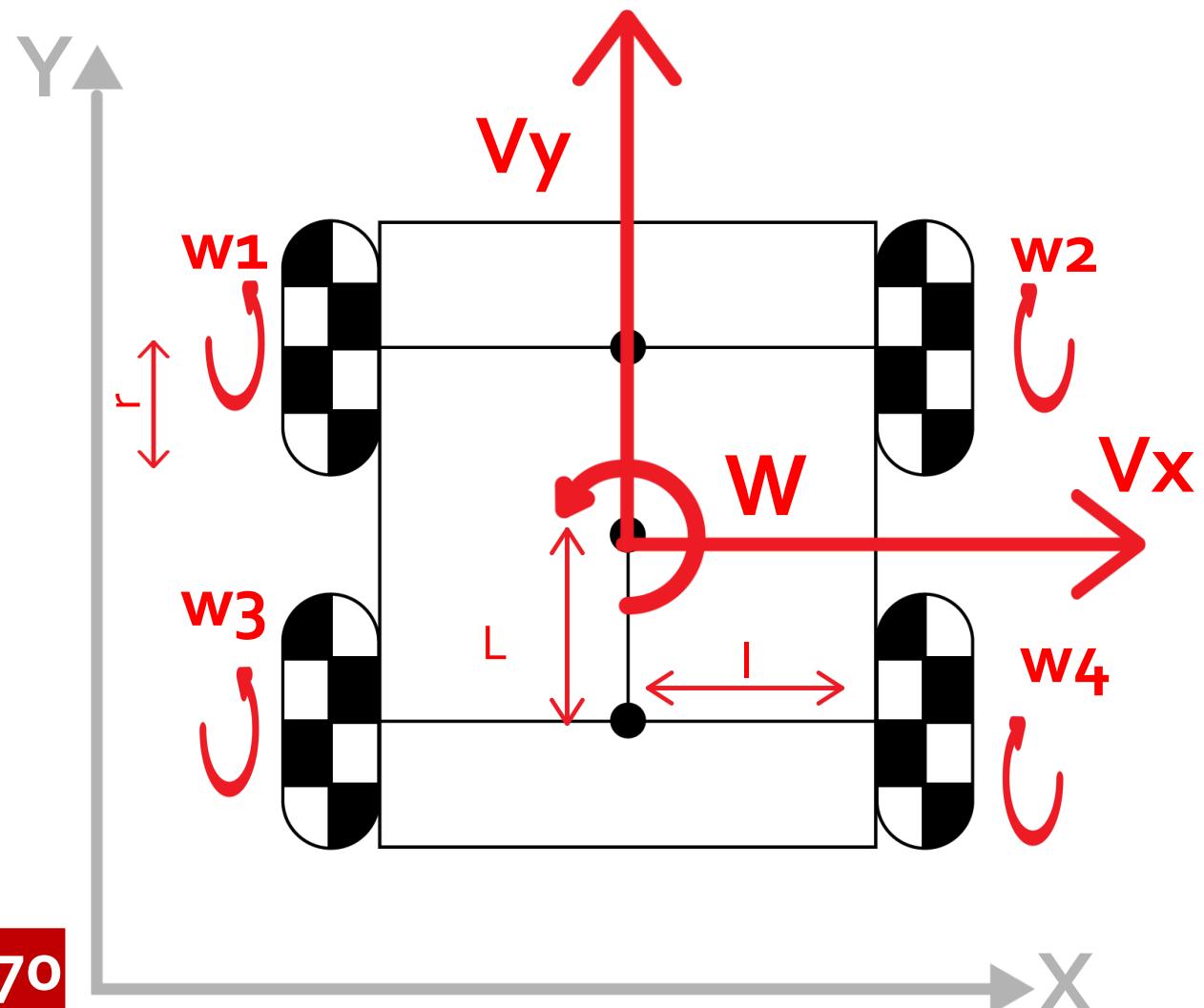


$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \left(\frac{1}{r}\right) \begin{bmatrix} 1 & 1 & - (L + l) \\ -1 & 1 & (L + l) \\ -1 & 1 & - (L + l) \\ 1 & 1 & (L + l) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}$$

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

WMR 4 LLANTAS: 4 OMNIDIRECCIONALES



$$w_1 = \frac{1}{r} (v_x + v_y - (L + l)w)$$
$$w_2 = \frac{1}{r} (-v_x + v_y + (L + l)w)$$
$$w_3 = \frac{1}{r} (-v_x + v_y - (L + l)w)$$
$$w_4 = \frac{1}{r} (v_x + v_y + (L + l)w)$$

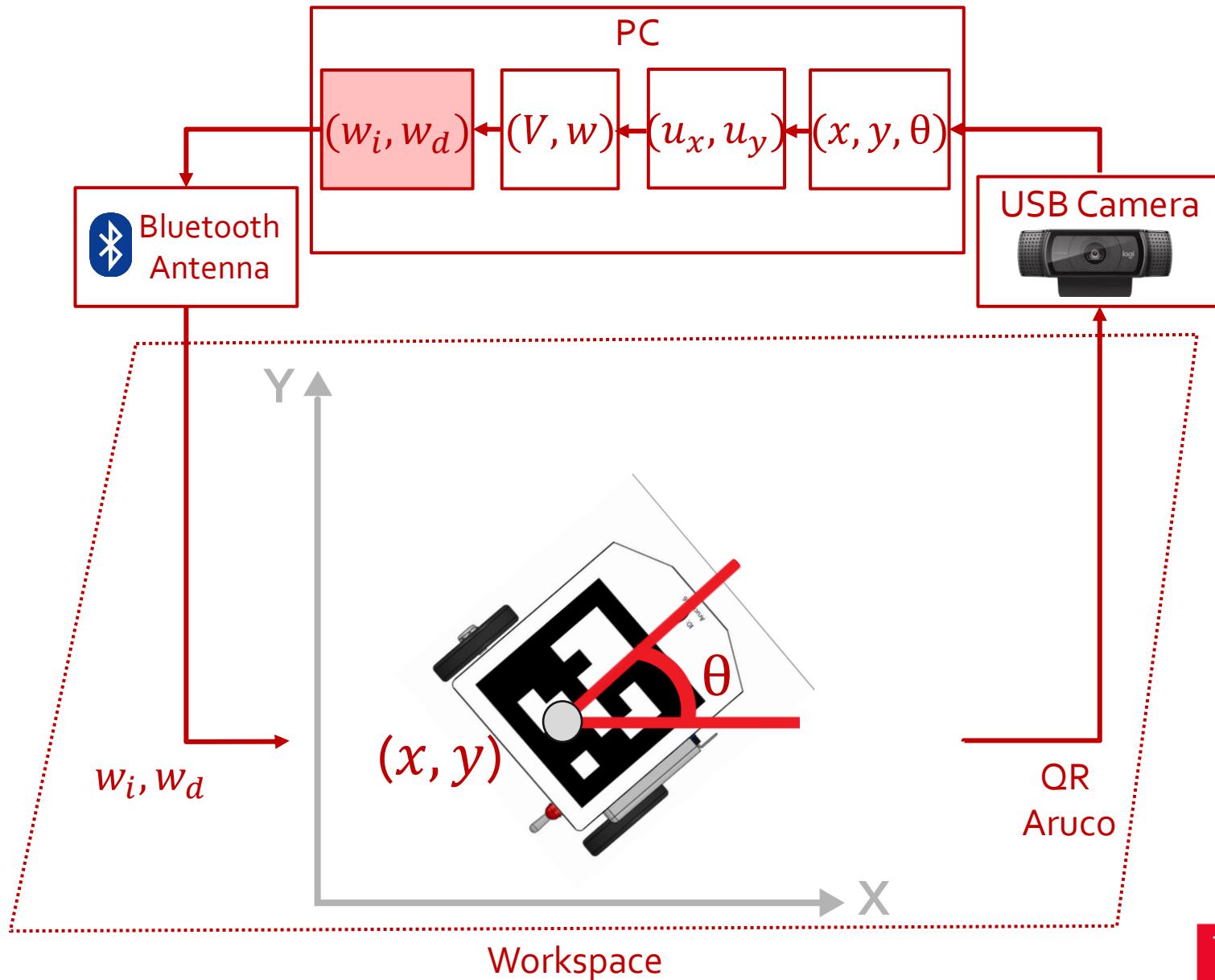


Figura 5.4. Plataforma de experimentación.

6. EXTENSIÓN DINÁMICA

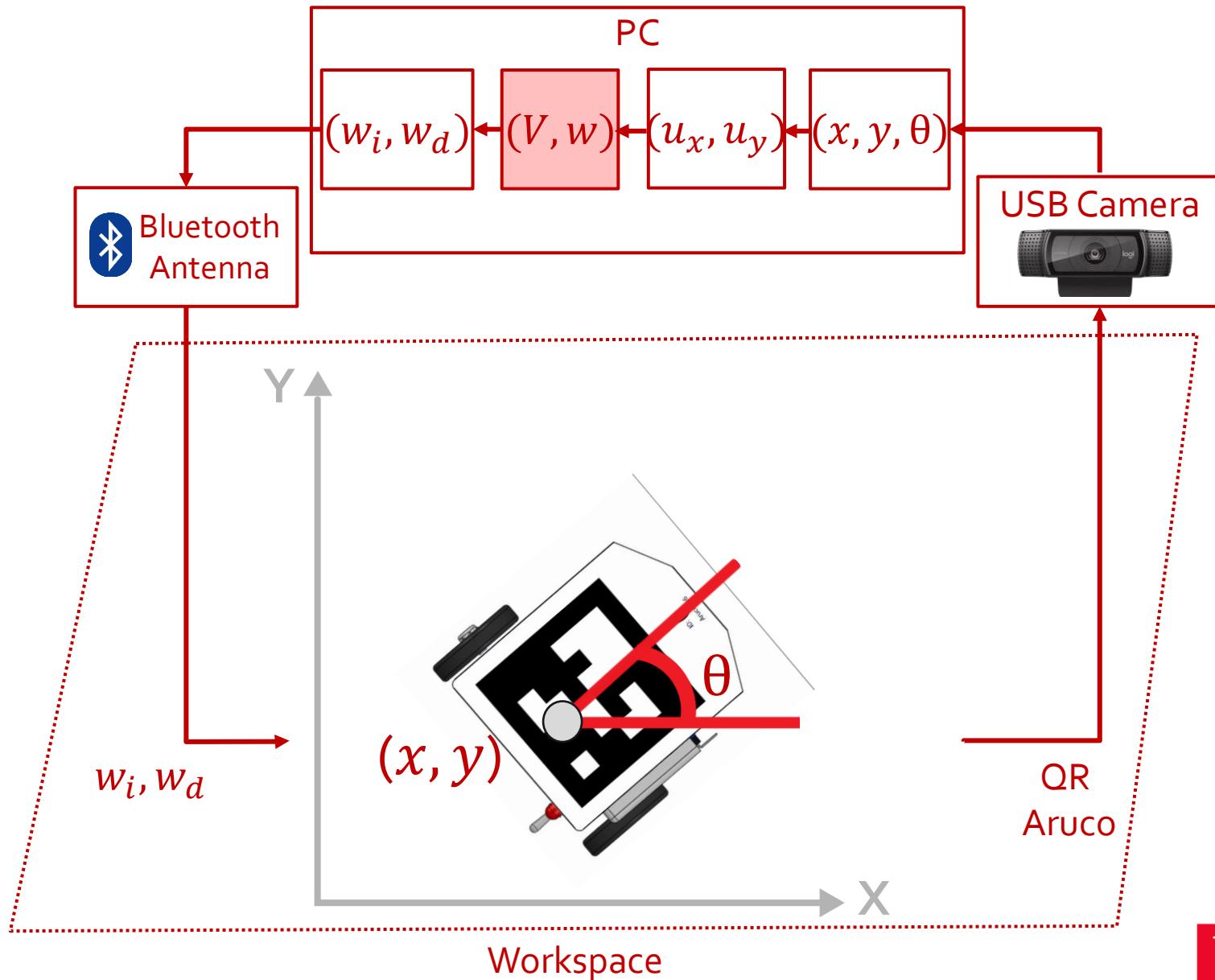
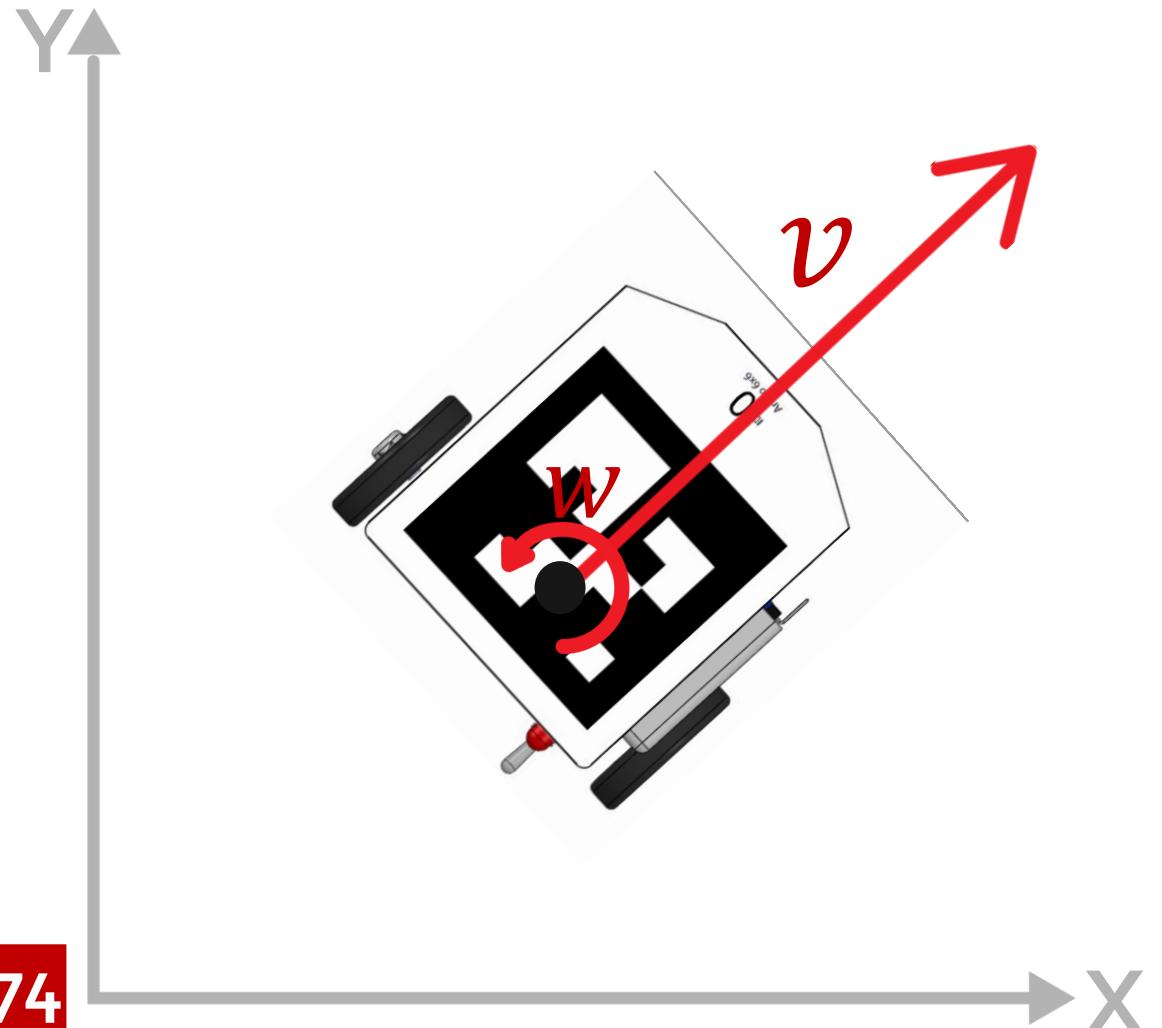


Figura 6.1. Plataforma de experimentación.



(x, y) → posición actual

v → Velocidad Lineal

w → Velocidad Angular

$$\dot{x} = v_x \quad \dot{y} = v_y$$

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\dot{x} = v_x \quad \dot{y} = v_y$$

$$v_x = v \cos \theta$$

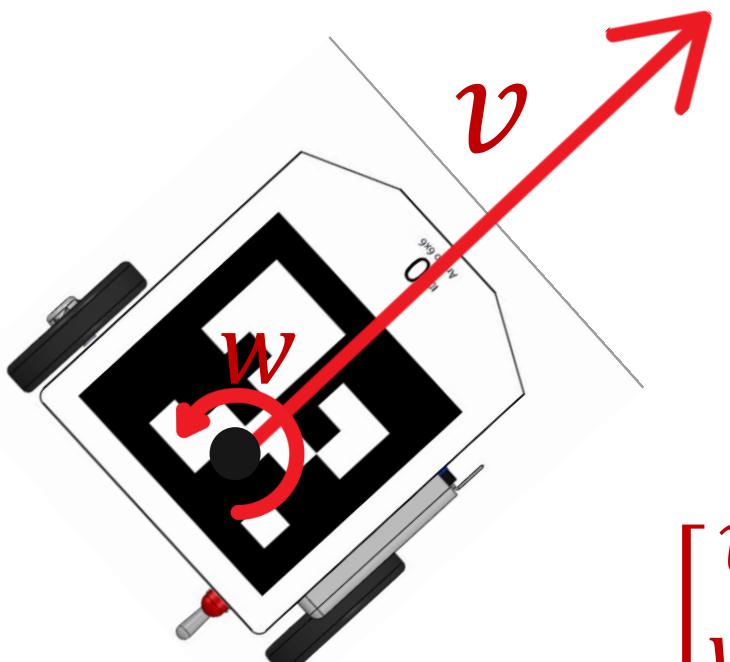
$$v_y = v \sin \theta$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

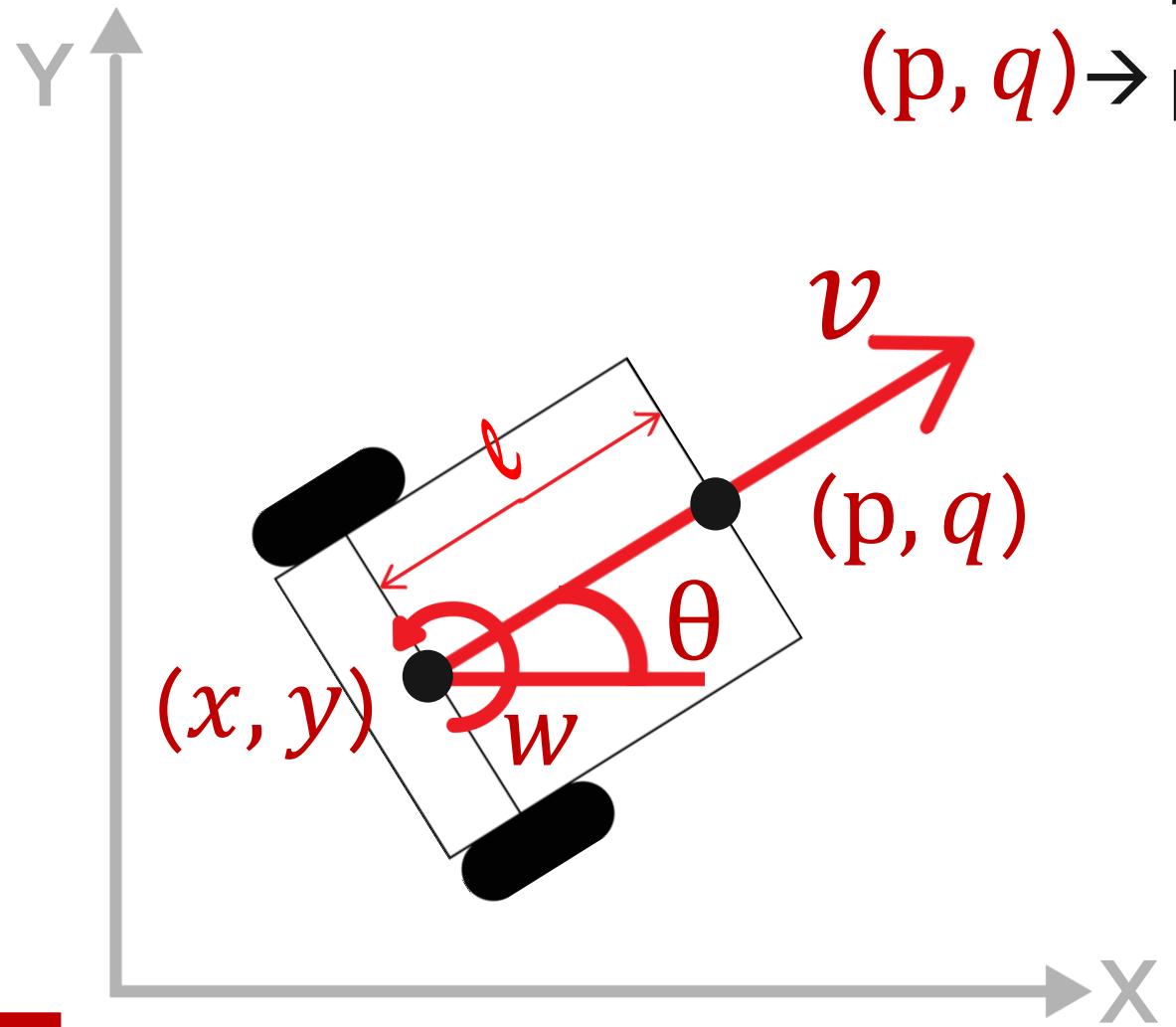
$$A = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix}$$

$A^{-1} = No existe$

$$\begin{bmatrix} v \\ w \end{bmatrix} = A^{-1} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$



EXTENSIÓN



$(x, y) \rightarrow$ posición actual

$(p, q) \rightarrow$ punto frontal

$$\begin{aligned}\dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= w\end{aligned}$$

$$p = x + \ell \cos \theta$$

$$q = y + \ell \sin \theta$$

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

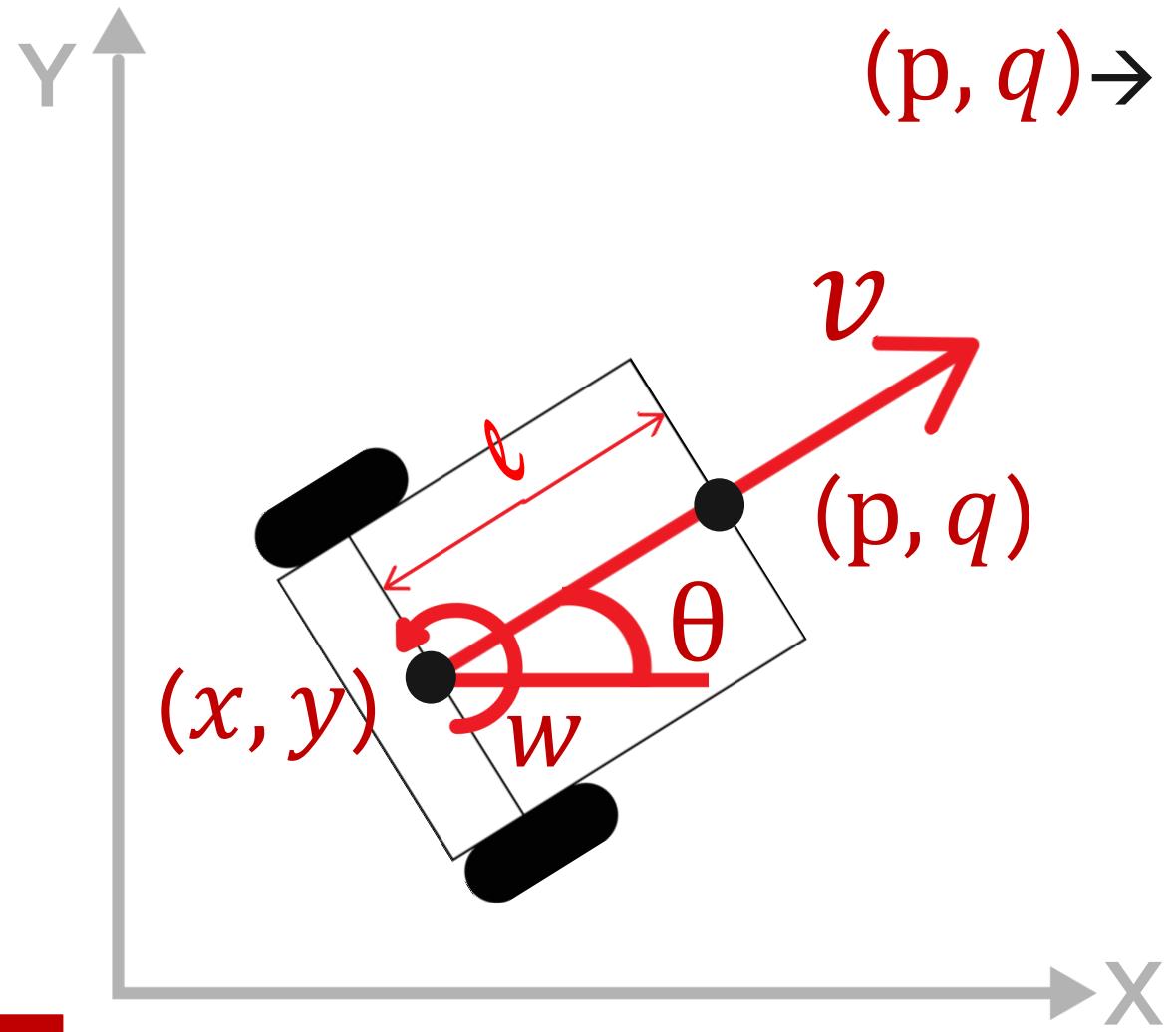
$$\dot{p} = \dot{x} - (\ell \sin \theta) * \dot{\theta}$$

$$\dot{p} = v \cos \theta - (\ell \sin \theta) * w$$

$$\dot{q} = \dot{y} + (\ell \cos \theta) * \dot{\theta}$$

$$\dot{q} = v \sin \theta + (\ell \cos \theta) * w$$

EXTENSIÓN



$(x, y) \rightarrow$ posición actual

$(p, q) \rightarrow$ punto frontal

$$\begin{aligned}\dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= w\end{aligned}$$

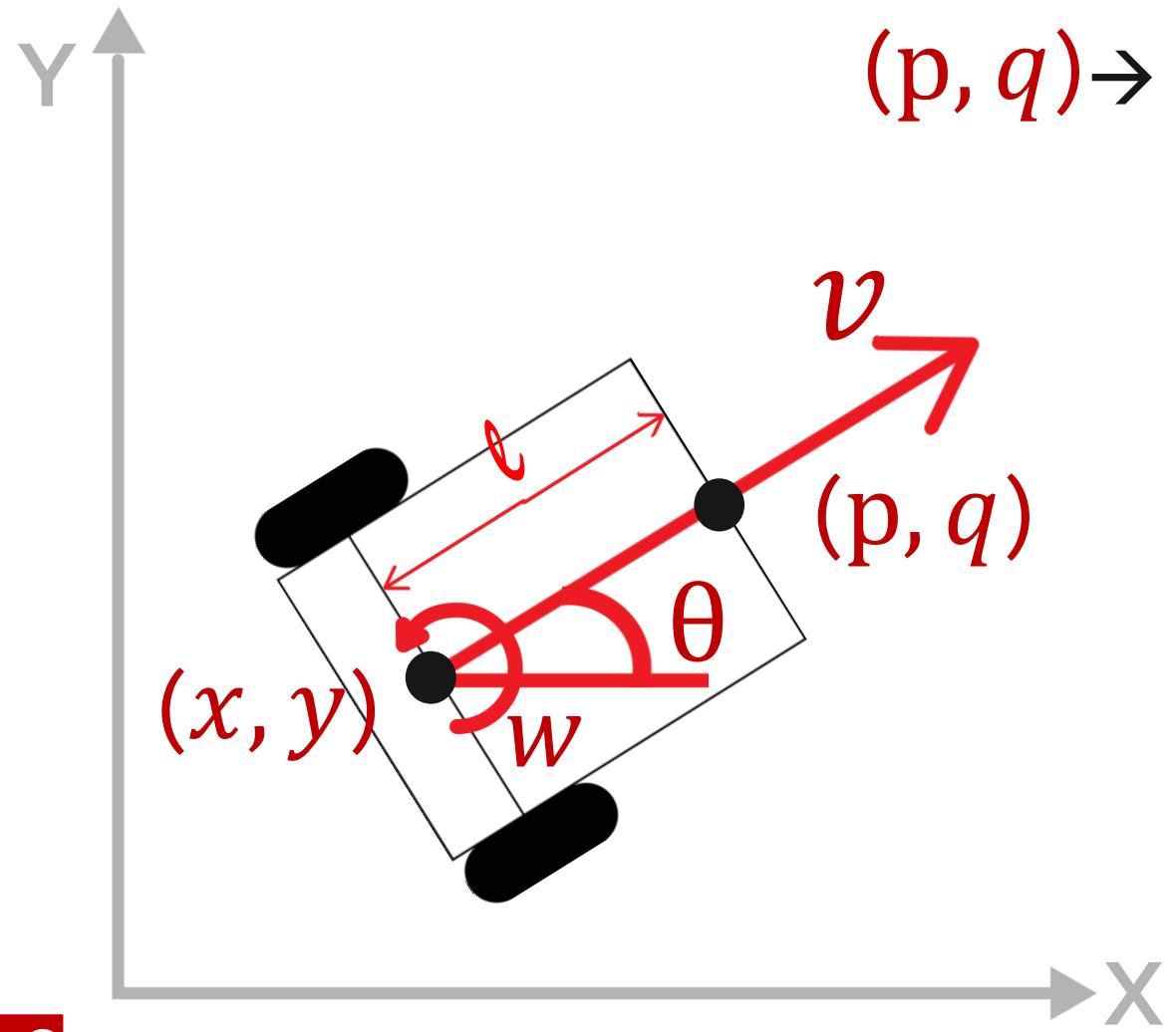
$$\dot{p} = v \cos \theta - (\ell \sin \theta) * w$$

$$\dot{q} = v \sin \theta + (\ell \cos \theta) * w$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

$$A = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix}$$

EXTENSIÓN



$(x, y) \rightarrow$ posición actual

$(p, q) \rightarrow$ punto frontal

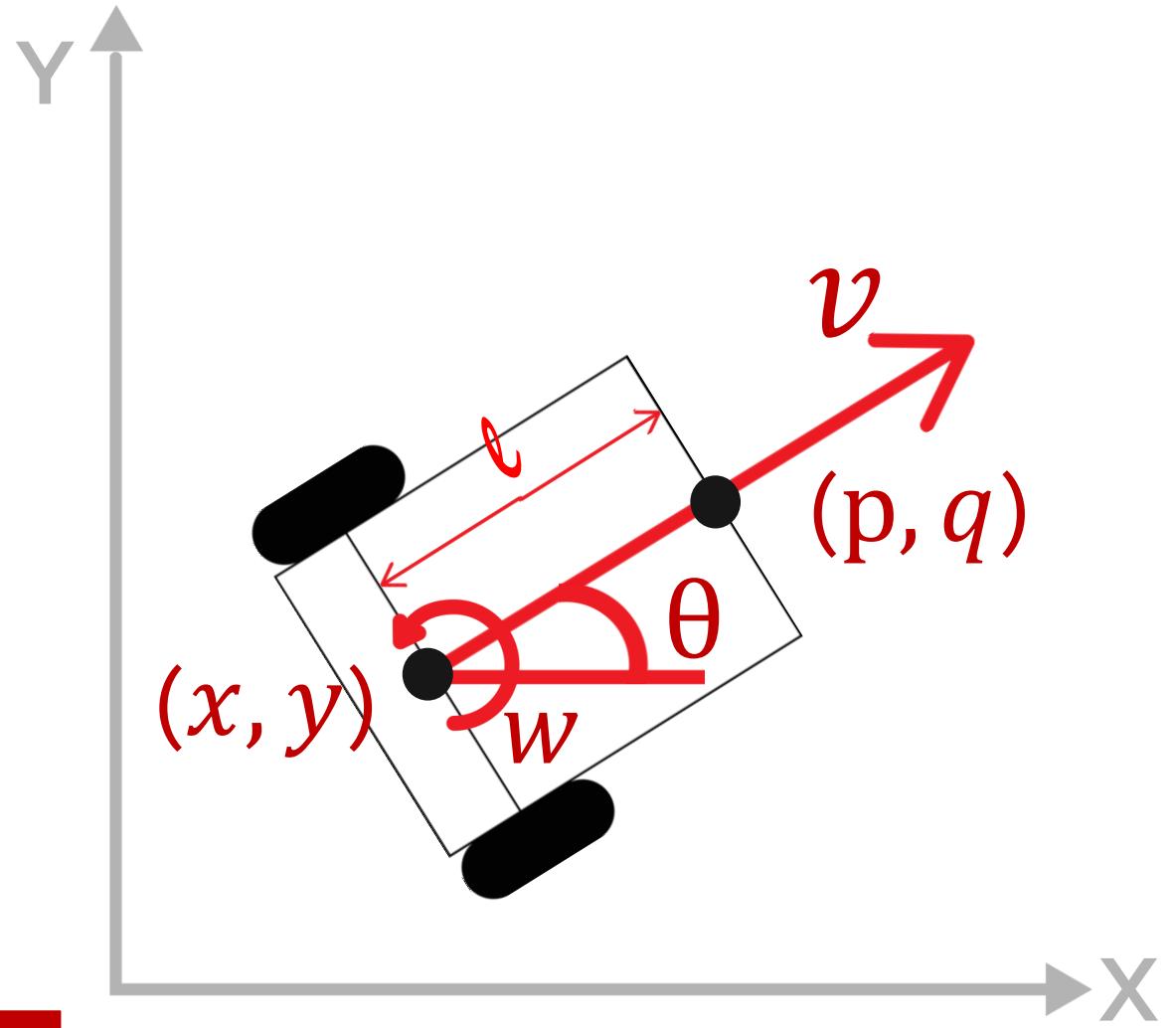
$$\begin{aligned}\dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= w\end{aligned}$$

$$A = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = A^{-1} \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix}$$

$A^{-1} = Si \ existe$

EXTENSIÓN



$(x, y) \rightarrow$ posición actual
 $(p, q) \rightarrow$ punto frontal

Si $\ell \approx 0$
entonces:

$$(x, y) \approx (p, q)$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} \approx \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix}$$

7. ESTRATEGIA DE CONTROL: ROBOT UNICICLO.

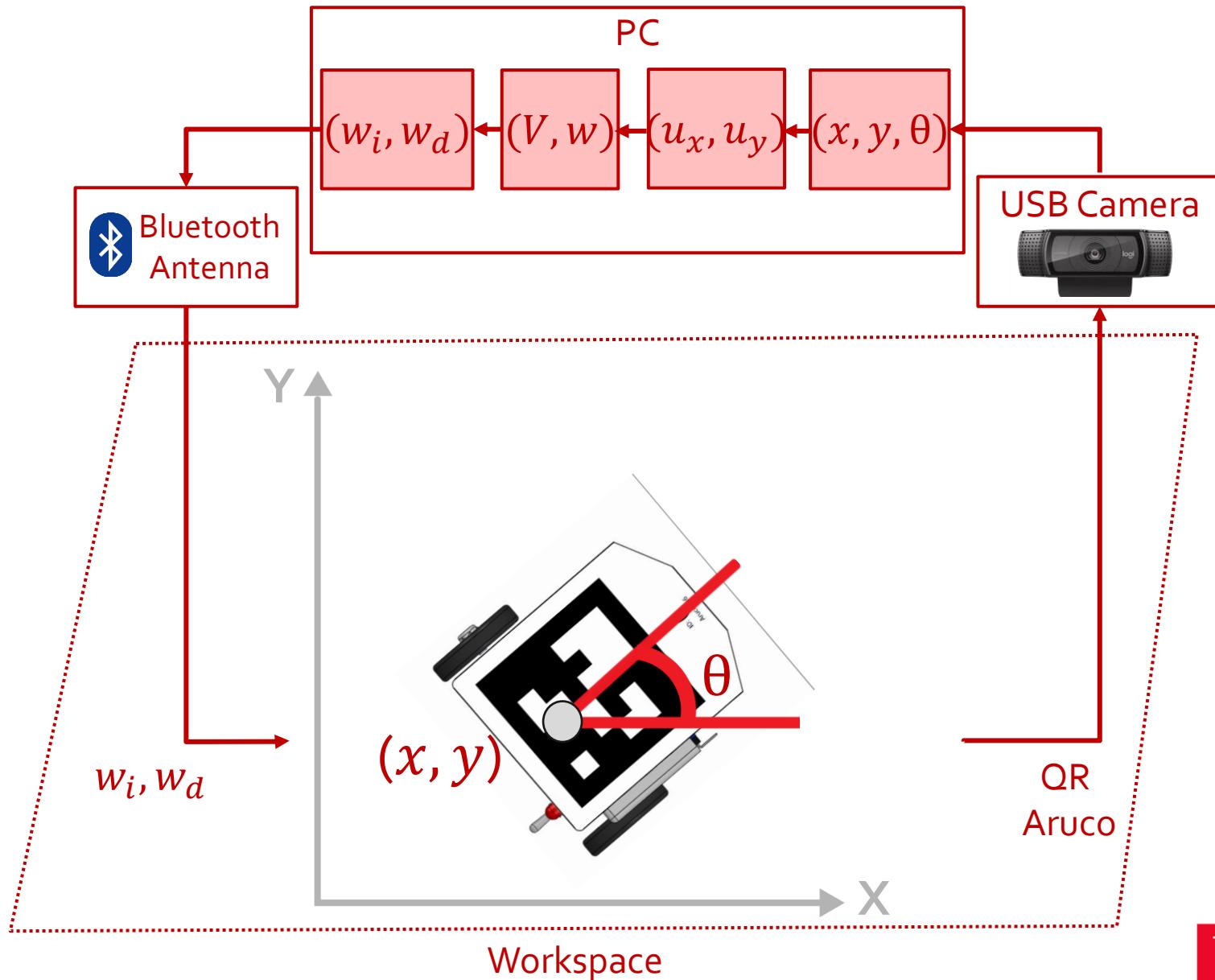
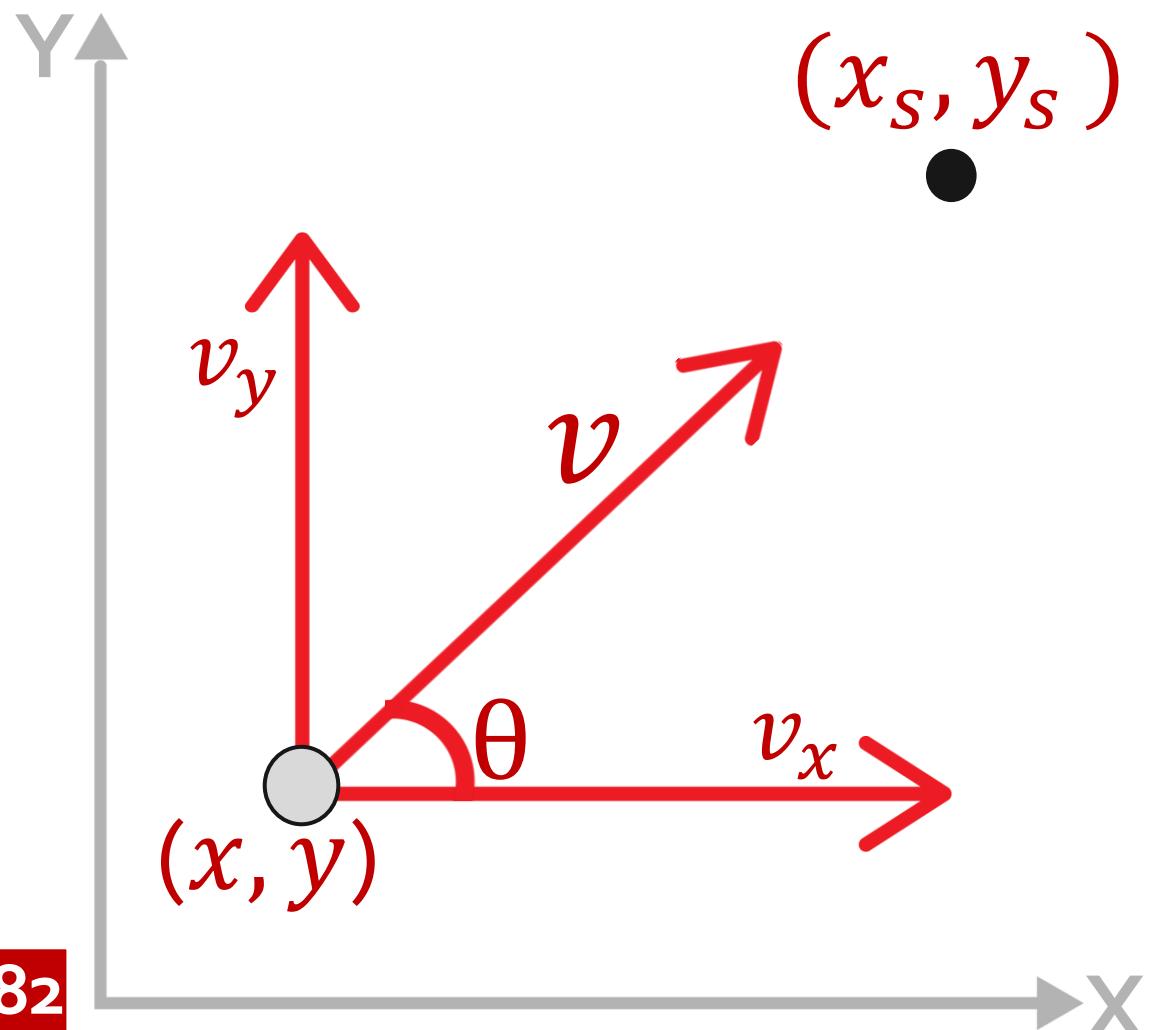


Figura 7.1. Plataforma de experimentación.

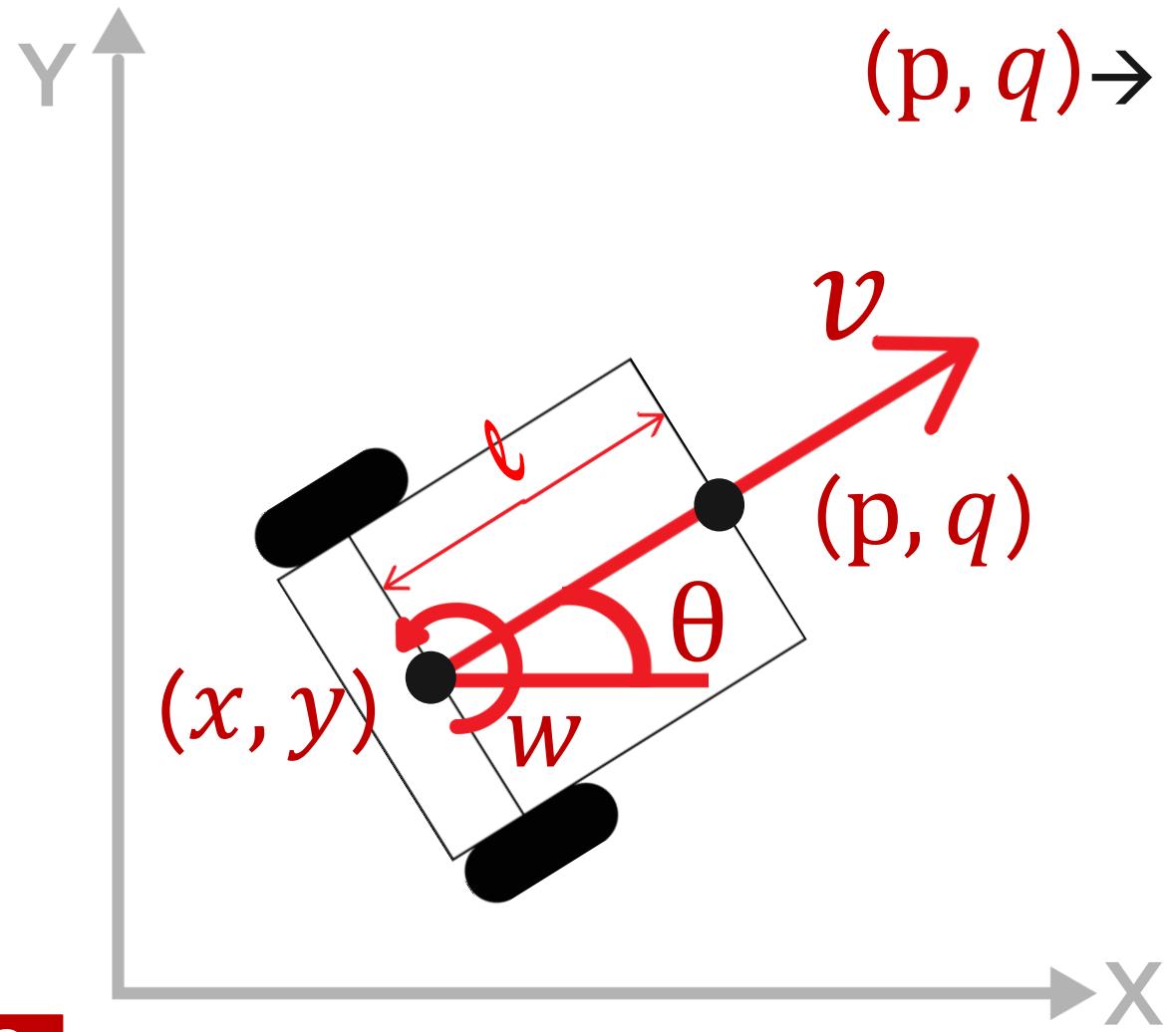
CONVERGENCIA



$\theta \rightarrow$ orientación
 $(x, y) \rightarrow$ posición actual
 $(x_s, y_s) \rightarrow$ posición deseada
(star point)

$$k = 1$$
$$error_x = x - x_s$$
$$error_y = y - y_s$$
$$u_x = -k * error_x$$
$$u_y = -k * error_y$$

EXTENSIÓN



$(x, y) \rightarrow$ posición actual

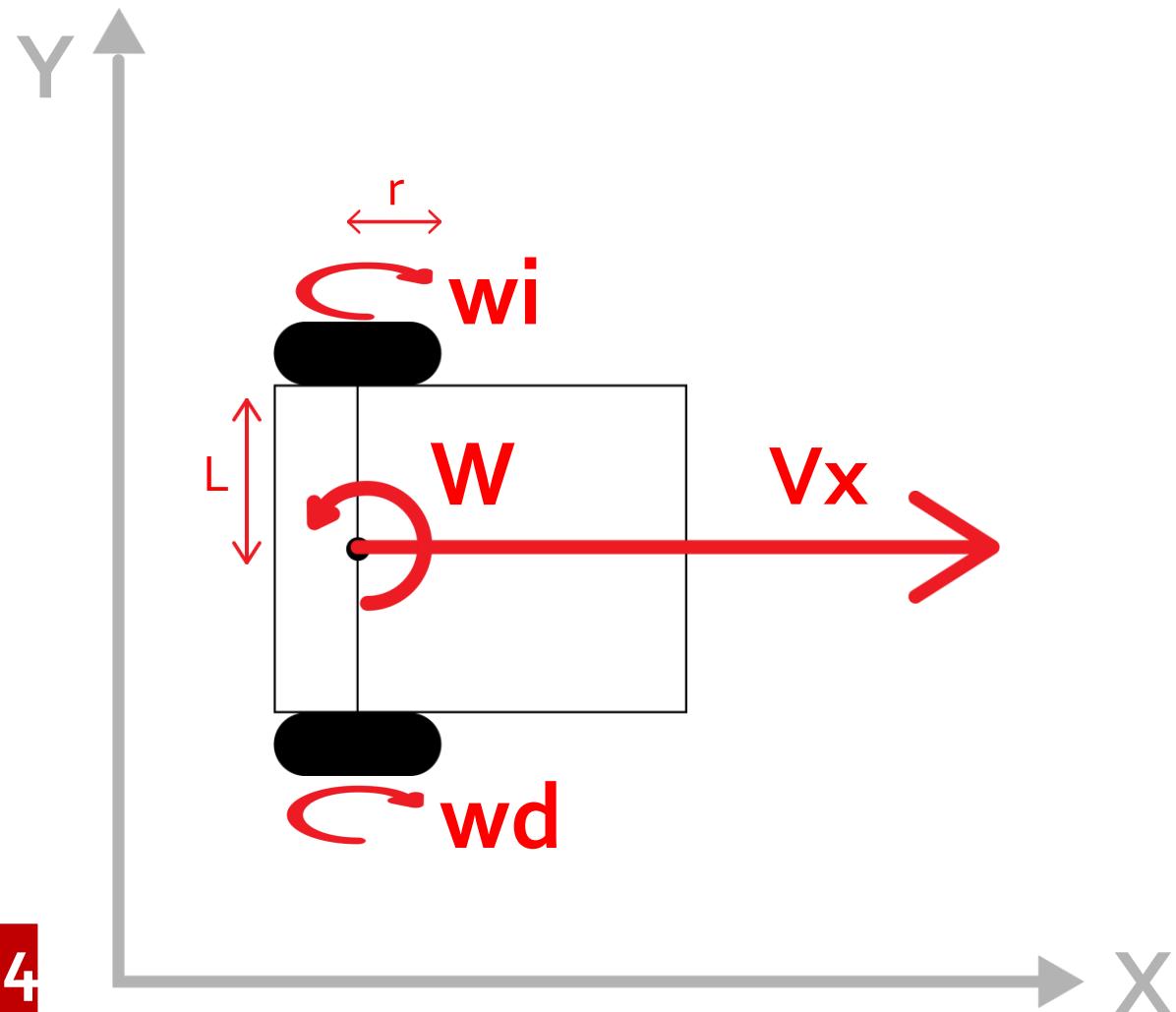
$(p, q) \rightarrow$ punto frontal

$$\begin{aligned}\dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= w\end{aligned}$$

$$A = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = A^{-1} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

WMR 2 LLANTAS: 2 FIJAS



$$\begin{bmatrix} w_i \\ w_d \end{bmatrix} = \left(\frac{1}{r} \right) \begin{bmatrix} 1 & -L \\ 1 & L \end{bmatrix} \begin{bmatrix} v_x \\ w \end{bmatrix}$$

$$w_i = \frac{1}{r} (v_x - Lw)$$

$$w_d = \frac{1}{r} (v_x + Lw)$$

CÓDIGO PYTHON

```
-----BUSQUEDA DE QR´s-----
#print("Buscamos aruco")
frame, points, ids =cam.buscar_Aruco(camara, resolucionx, resoluciony)
#print(ids)

if len(points)>0:
    robot=cam.buscar_robots(points, ids, robot)
    #print("robot0(x="+str(robot[0][0])+","y="+str(robot[0][1])+",th="+str

-----LEY DE CONTROL CONVERGENCIA-----

x=robot[robot_id][0]-(resolucionx/2) #Obtenemos valor X de QR robot
y=robot[robot_id][1]-(resoluciony/2) #Obtenemos valor Y de QR robot
th=robot[robot_id][2] #Obtenemos valor Th de robot
```

CÓDIGO PYTHON

- Cálculo de: extensión en punto p, q

```
x=x+(l*np.cos(th))
y=y+(l*np.sin(th))

A=np.array([[np.cos(th), -l*np.sin(th)],
            [np.sin(th), l*np.cos(th)]])

ex,ey,ux,uy=robotx.convergencia(x,y,xs,ys,k)
```

- Cálculo de: error, ux, uy

```
def convergencia(x,y,xs,ys,k):
    #Calculo de error
    ex=x-xs
    ey=y-ys

    #Control proporcional
    ux=-k*ex
    uy=-k*ey

    return ex,ey,ux,uy
```

CÓDIGO PYTHON

- Cálculo de: V, w

```
B=np.array([ux,uy]) #Arreglo de Vector velocidad  
  
U=np.linalg.solve(A,B)  
#U=np.linalg.inv(A)*B No sirve  
  
V=U[0] #Velocidad Lineal  
W=U[1] #Velocidad Angular
```

CÓDIGO PYTHON

- Cálculo de wd, wi

```
#-----MODELO CINEMATICO ROBOT-----  
wd= (V/r)+((L*w)/(2*r)) #Calulo de wd robot uniciclo  
wi= (V/r)-((L*w)/(2*r)) #Calulo de wi robot uniciclo
```

```
if(wd>wmax):  
    wd=wmax  
elif(wd<-wmax):  
    wd=-wmax  
  
if(wi>wmax):  
    wi=wmax  
elif(wi<-wmax):  
    wi=-wmax
```

- Envio de Bluetooth:

```
Bt.move(robot_bt,wd, wi)
```

CONVERGENCIA

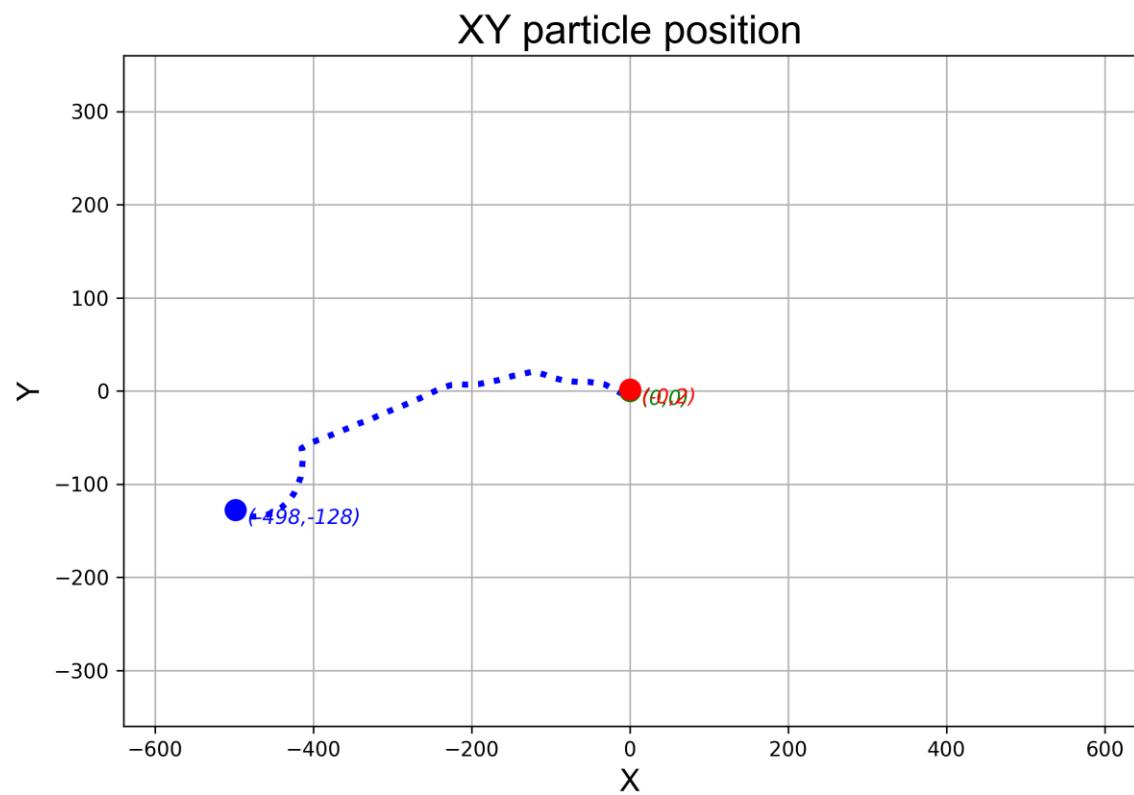


Figura 7.2. Grafica posición XY.

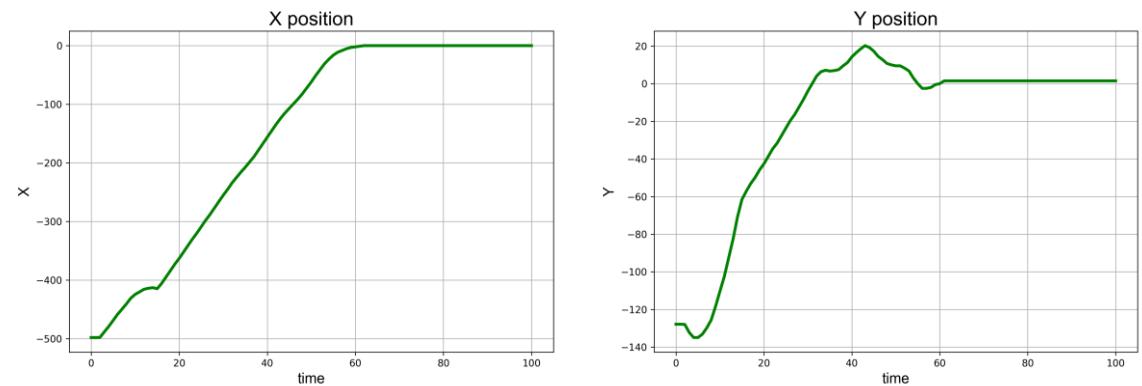


Figura 7.3. Graficas de posición.

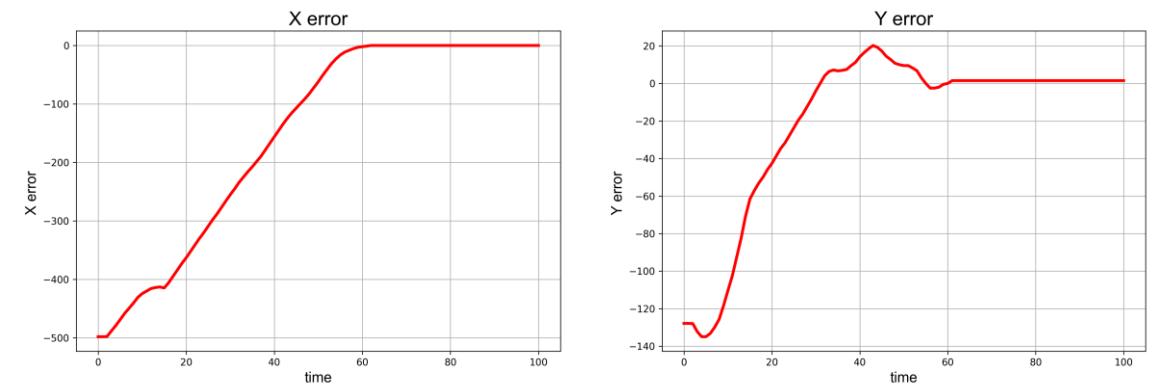
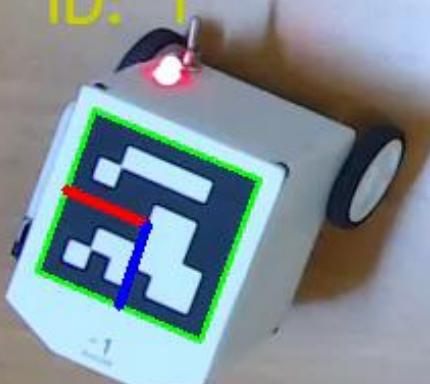


Figura 7.4. Graficas de error.

RC CONTROL (detener)

[-523.0 ,205.0]

ID: 1



XS,YS
0.0,0.0

TRAYECTORIA

XY particle position

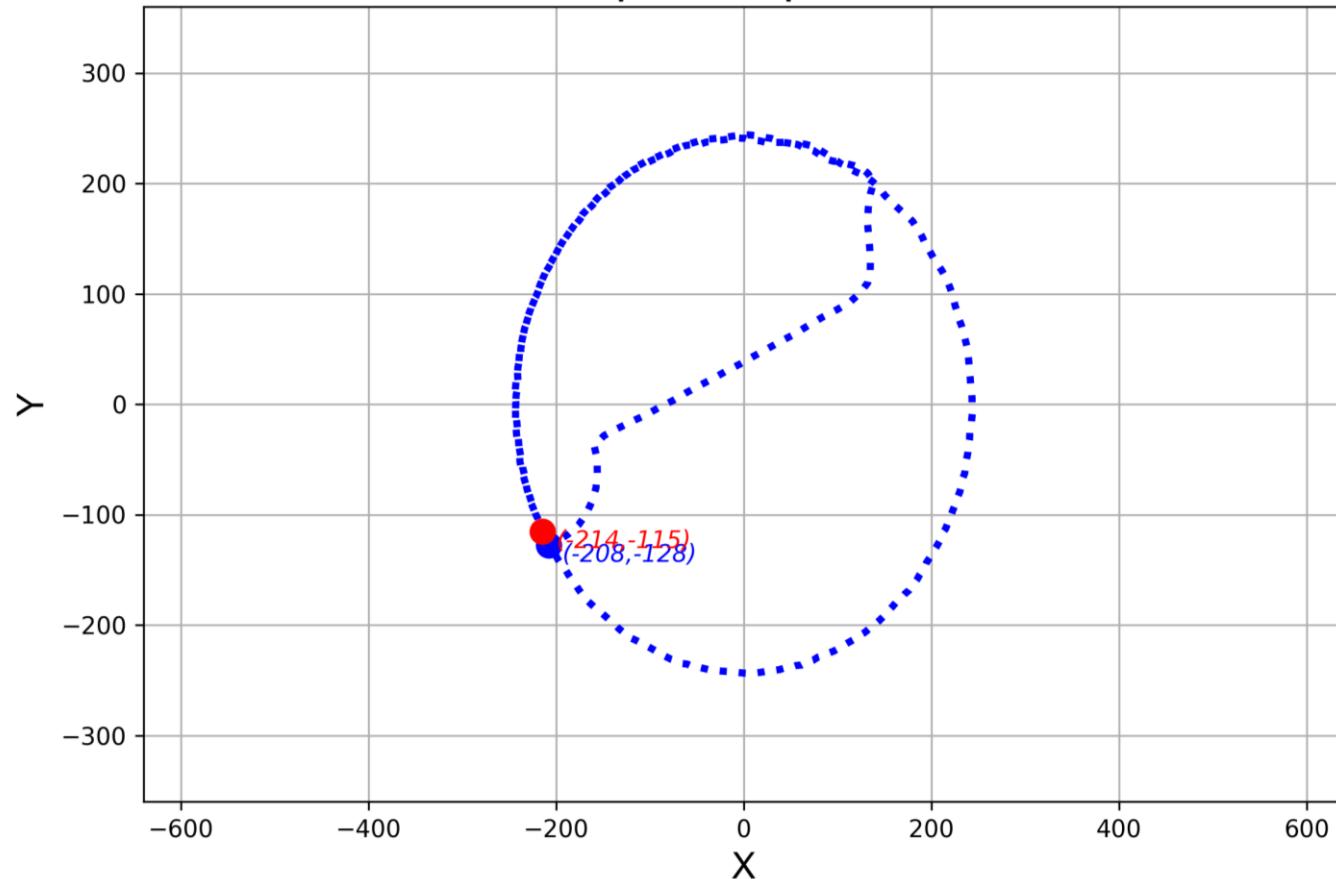


Figura 7.5. Grafica posición XY.

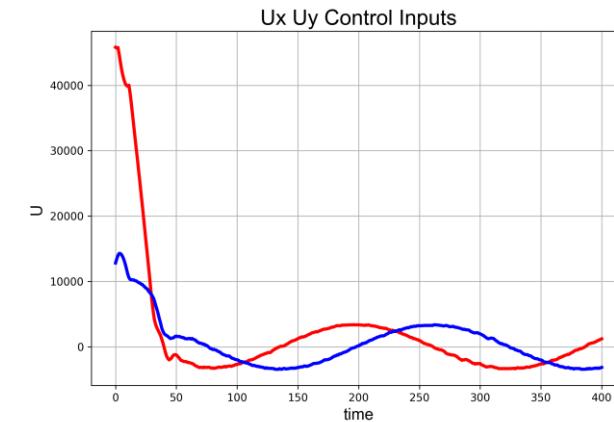


Figura 7.6. Graficas de ganancia en control.

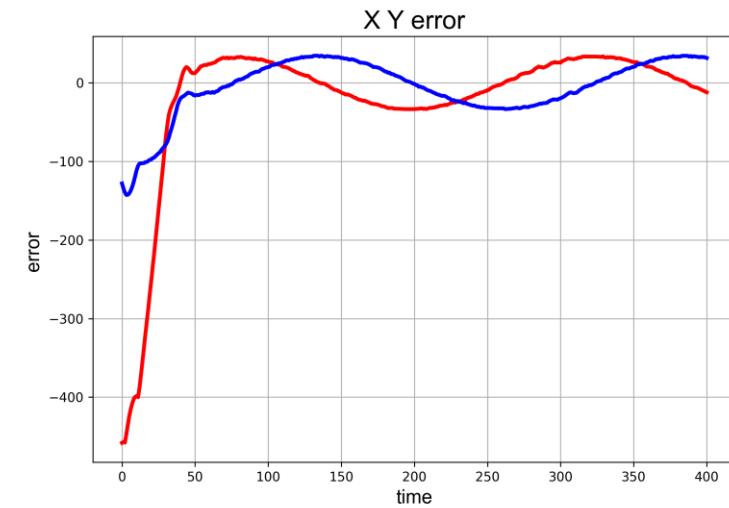
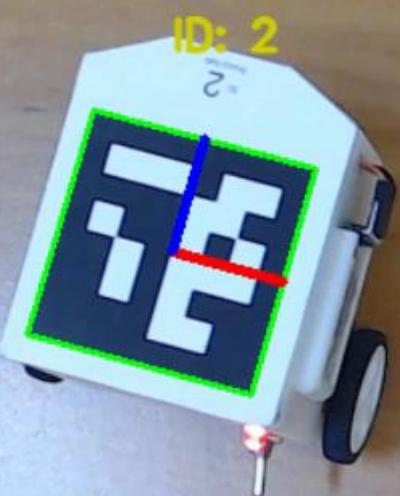


Figura 7.7. Graficas de error.

RC CONTROL (detener)

[-214.0 , -104.0]



ID: 2

FORMACIÓN

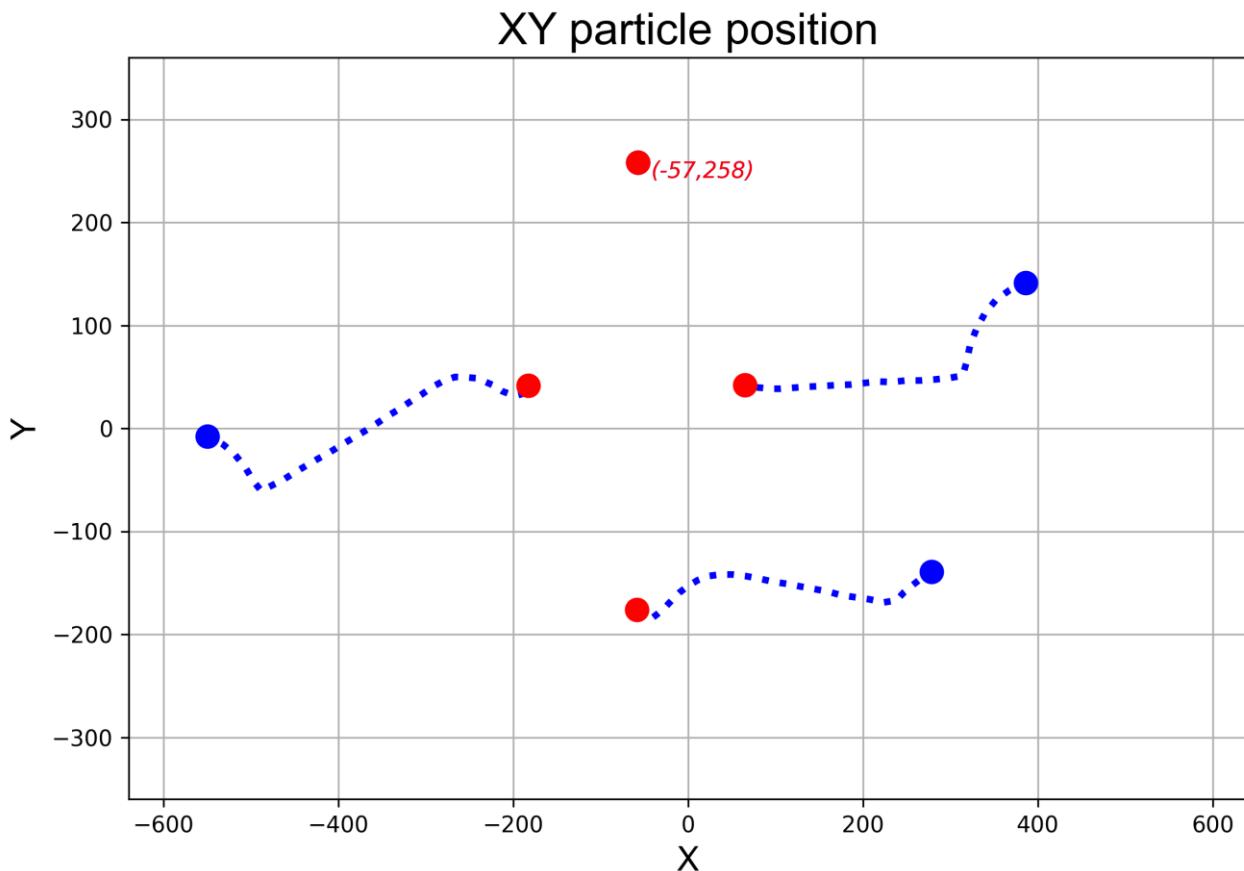


Figura 7.8. Grafica posición XY.

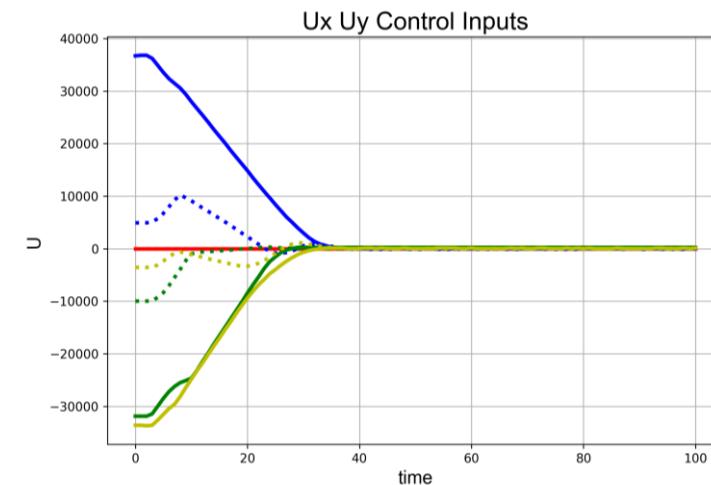


Figura 7.9. Graficas de ganancia en control.

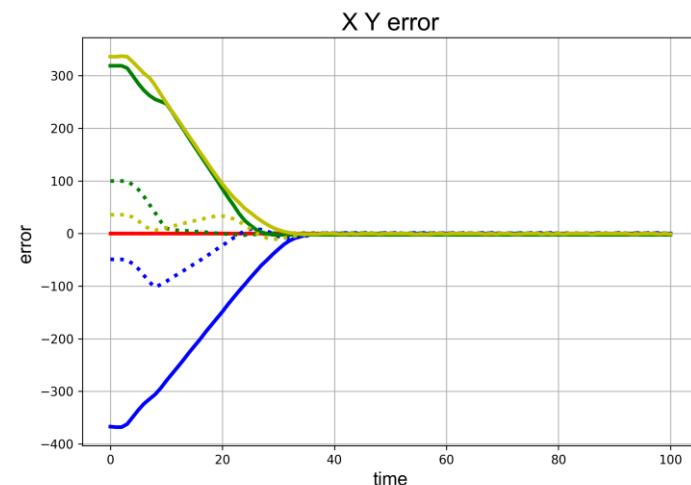
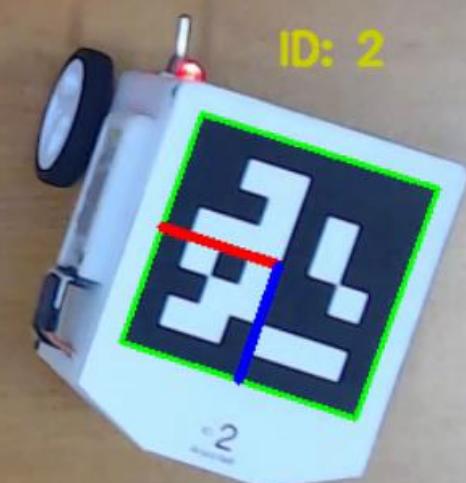
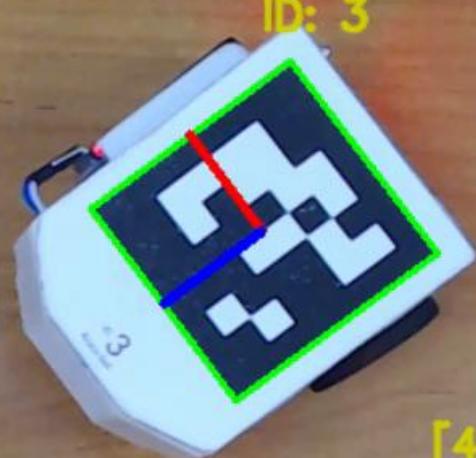
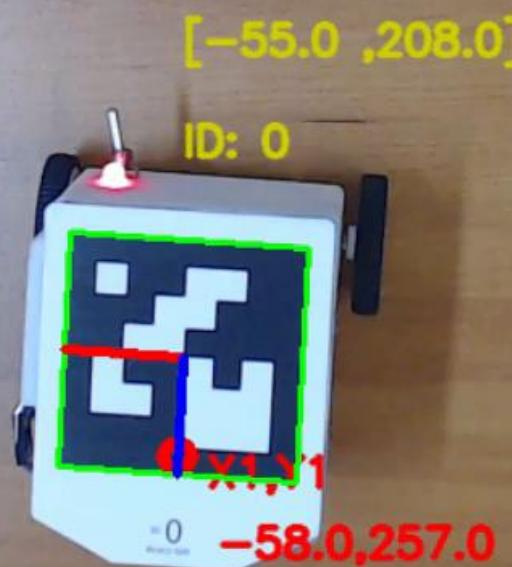


Figura 7.10. Graficas de error.

RC CONTROL (detener)



FORMACIÓN - TRAYECTORIA

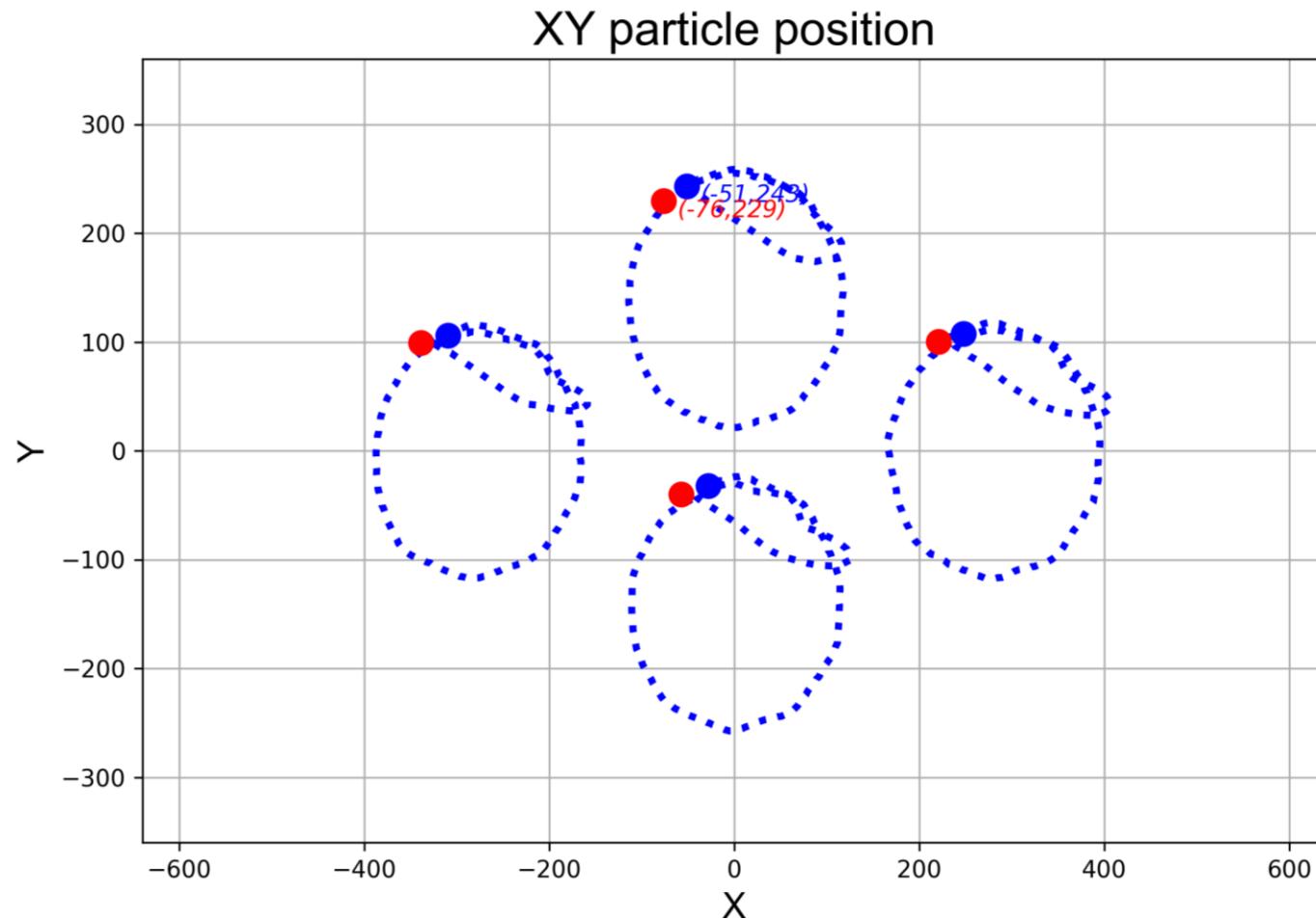


Figura 7.11. Grafica posición XY.

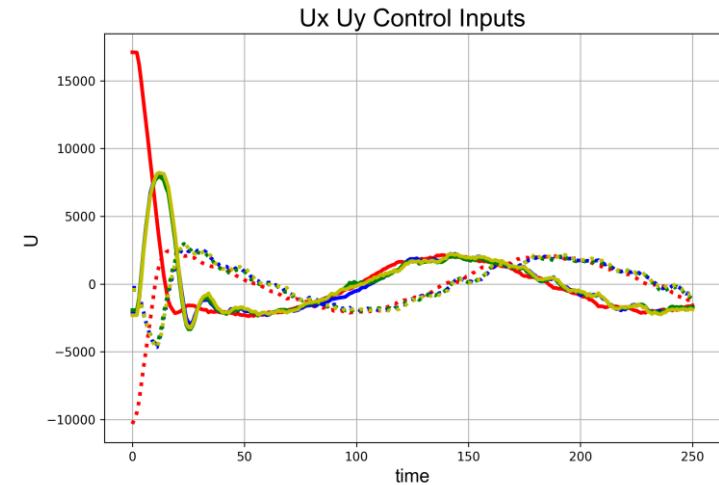


Figura 7.12. Graficas de ganancia en control.

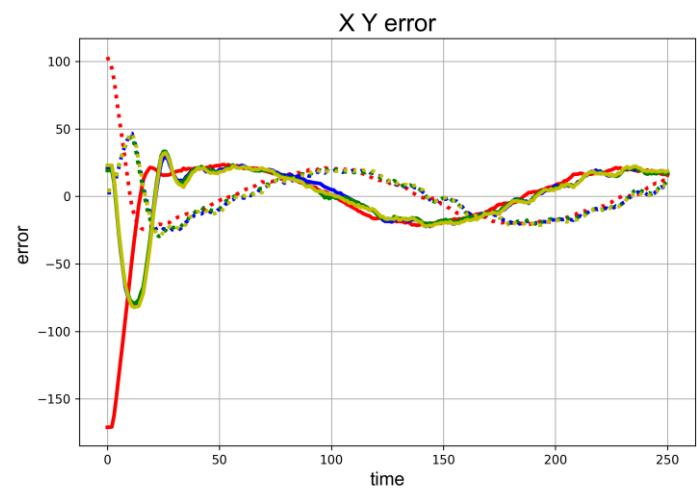
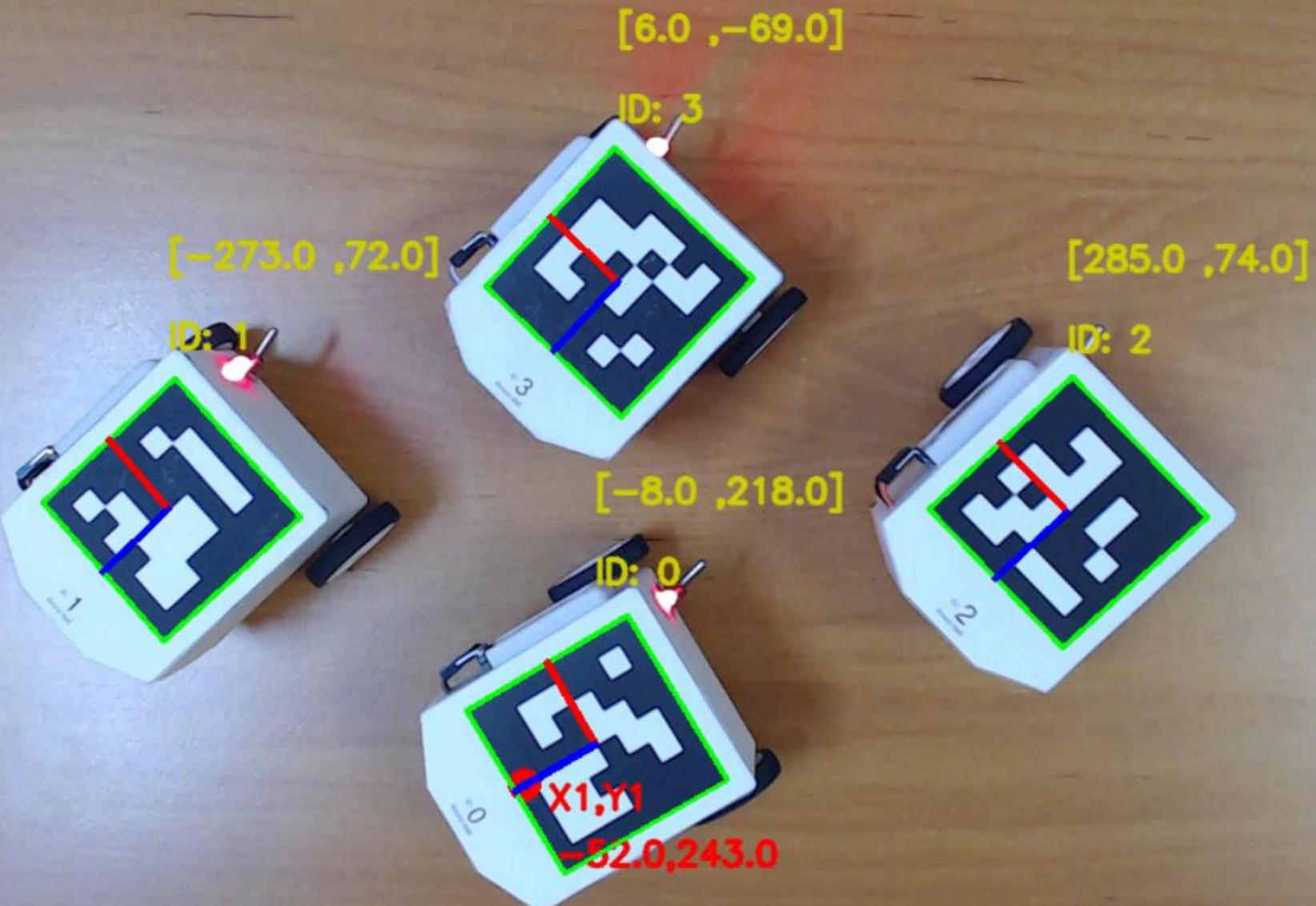


Figura 7.13. Graficas de error.

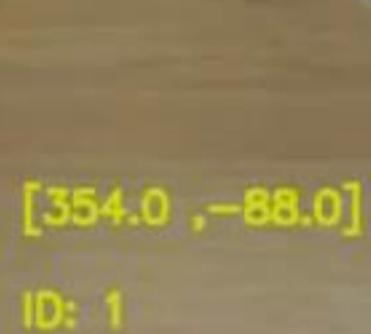
RC CONTROL (detener)



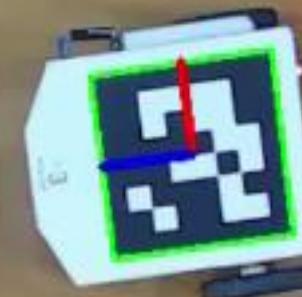
RC CONTROL (detener)



ID: 3



ID: 1



ID: 2



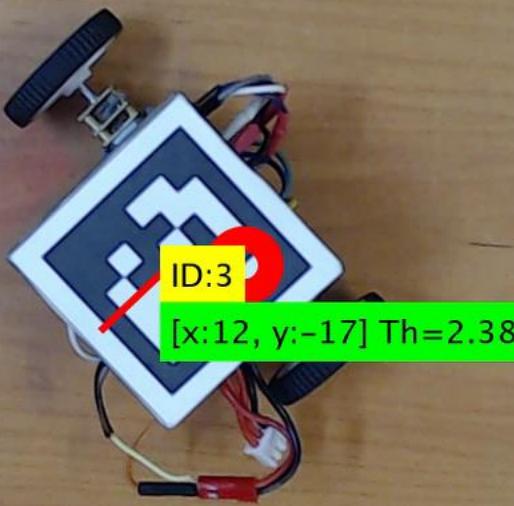
8. PROYECTO: ROBOT MÓVIL

PROYECTO FINAL

- Desarrollar un robot móvil con alguna de las configuraciones mostradas en la sección de modelos cinemáticos.
- Desarrollar cada uno de los siguientes puntos:
 1. Desarrollo de robot móvil con Arduino
 2. Implementar el modelo cinemático del robot con Python y Arduino que permita su control remoto.
 3. Implementar el sistema de visión por computadora con códigos Aruco.
 4. Implementa las estrategias de control de convergencia y trayectoria.
 5. Implementa con el apoyo de otros 2 equipos y sus robots, la estrategia de formación para formar un triángulo.
 6. EXTRA: Implementa formación y trayectoria con 3 robots.



Figura 1.1. Robots móviles con ruedas.



REFERENCIAS

- [1] Carlos Canudas de Wit. (1996). Theory of Robot Control. Londres: Springer.
- [2] Siegwart, R. (2004). Introduction to autonomous mobile robots. Cambridge: Massachusetts Institute of Technology Press.
- [3] Tzafestas, S. G. (2014). Introduction to Mobile Robot Control. Atenas: Elsevier.
- [4] Peter Corke. (2017). Robotics, Vision and Control, Funtamental Algorithms in MATLAB. Australia: Springer.