



UNIVERSIDAD DE MONTERREY
DIVISIÓN DE INGENIERÍA Y TECNOLOGÍAS
DEPARTAMENTO DE INGENIERÍA
PROGRAMA DE EVALUACIÓN FINAL

VEHÍCULO AUTÓNOMO GUIADO APLICADO A TAREAS DE INSPECCIÓN Y ENTREGA

Asesor: Jorge de Jesús Lozoya Santos, Ph.D.

Alcalá, Osmar

De la Mora, Didier

Treviño, Hugo

Diciembre 1, 2017

Agradecimientos

Agradecemos a nuestro asesor Dr. Jorge Lozoya por ser guía y apoyo durante todo el proyecto, al Dr. Arturo E. Cerón y Cesár Hernández que, a pesar de no encontrarse en el país, siempre estuvieron en contacto con el equipo para el desarrollo del proyecto, al profesor Patricio Ortiz Silva por su compromiso y colaboración con el diseño del mismo.

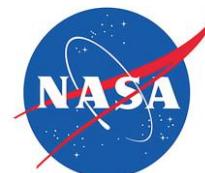
Contenido

- Introducción
- Estado del arte
- Objetivo
- Desarrollo/Metodología
- Resultados
- Conclusiones

Introducción



Creciente investigación, desarrollo y aplicación
de los vehículos autónomos por parte de
compañías internacionales e instituciones
académicas de prestigio





iRobot® Roomba® 980



Waymo Chrysler Pacifica
Hybrid



NASA Mars Exploration Rover

Capacidad de explorar y navegar en ambientes dinámicos y/o desconocidos



Amazon Kiva



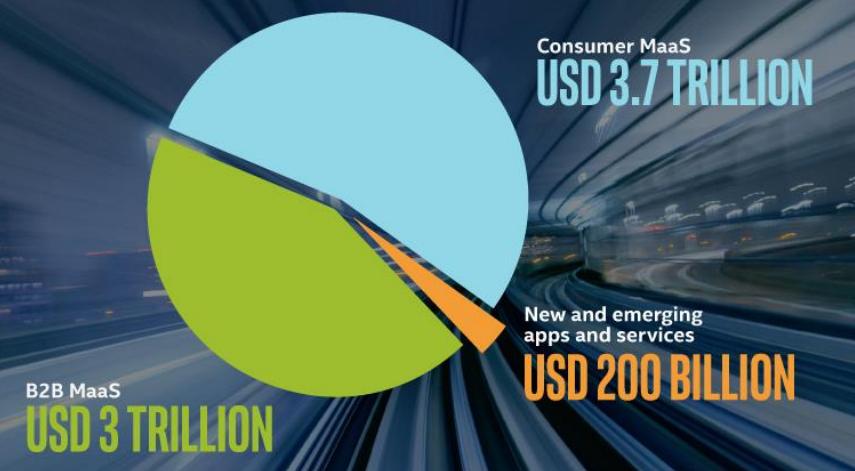
LoweBot NAVii™

Welcome to the Passenger Economy



In the near future, people everywhere will leave the driving to autonomous vehicles. That one simple fact will soon create one of the greatest economic opportunities of the twenty-first century.

THE USD 7 TRILLION FUTURE BY 2050



SOCIAL IMPACT

Autonomous vehicles will change human behavior forever.



PASSENGER FREEDOM

Riders can reduce stress and use travel time for more productive activities or interests.



CLEARER HORIZONS

With traffic orchestrated and optimized, the trip from Point A to Point B will happen faster, with less congestion on the way.



CURBSIDE APPEAL

The dramatic decrease in demand for parking will free urban spaces for more optimal uses.



MOBILE BEAUTY SALONS

for a blow-dry and manicure/pedicure on the way to work.



RESTAURANTS

that offer a free commute—for the price of a breakfast burrito.



MINI-CLINICS

providing basic exams en route to visit Mom.

A WHOLE NEW CUSTOMER-CENTRIC NORMAL

Uncoupled from the steering wheel, businesses will offer riders new, personalized choices on the go.



MOBILE MOTELS

for a quick power nap or an overnight snooze.



CONVENIENCE

stores or pharmacies that come to you.



DRIVE-AWAY MOVIES

customized for short and long travel times.

WHAT SHOULD YOU BE THINKING ABOUT?



CAR-VENIENCE VS. CONCIERGE SERVICE

The process of moving people will become a commodity. To win, businesses will need to cultivate customer relationships and expand the customer experience.



NEW IT REALITIES

While your labor costs may go down, IT needs will increase: predictive analytics, cloud services, and Internet of Things technologies will come into play.



DATA PROMOTED FROM KING TO EMPEROR

Data will become even more important as you use data analytics, predictive analytics, and system optimization to build customer loyalty through customized experiences.

ACHIEVING CRITICAL MAAS



The boundaries between work and personal life will continue to evolve and disappear.

MOBILE CONNECTEDNESS

+ By 2050, nearly two-thirds of the global population will live in an urban environment. (New suburban growth also means more commuters.)

URBANIZATION

+ Individual ownership is being replaced with a shared approach.

PASSENGER ECONOMY IN 2030 = USD 800 BILLION (estimated)

IT'S THE BIGGEST SHIFT IN HOW PEOPLE GET AROUND SINCE MODERN SOCIETY LEFT THE HORSE AND BUGGY BEHIND.

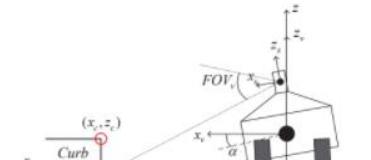
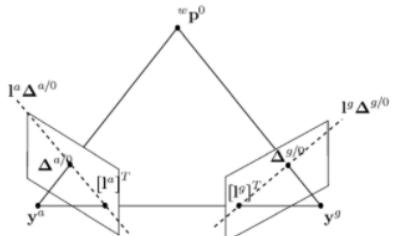
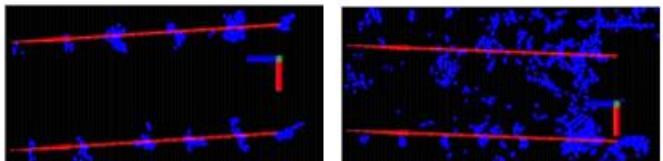
Intel. (1 de June de 2017). Intel Predicts Autonomous Driving Will Spur New 'Passenger Economy' Worth \$7 Trillion. Obtenido de Intel Newsroom: <https://newsroom.intel.com/news-releases/intel-predicts-autonomous-driving-will-spur-new-passenger-economy-worth-7-trillion/>



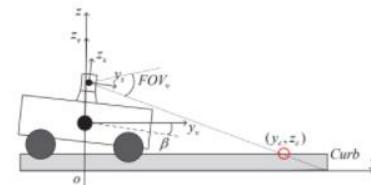
Estado del Arte

(artículos revisados: al menos 50, 29 seleccionados y analizados a detalle)

ScienceDirect



(a) Front view



"3D Perception for Accurate Row Following: Methodology and Results" Zhang, J., Chambers, A., Maeta, S., Bergerman, M., & Singh, S. (2013)

"Probabilistic Cooperative Target Localization " Nagaty, A., Thibault, C., Trentini, M., & Li, H. (2015)

"A Real-time Curb Detection and Tracking Method for UGVs by Using a 3DLIDAR Sensor" Zhang, Y., Wang, J., Wang, X., Li, C., & Wang, L. (2015)

"Integration of People Detection and Simultaneous Localization and Mapping Systems for an Autonomous Robotic Platform " Angonese, A. T., & Rosa, P. F. F. (2016)



UNIVERSIDAD
DE MONTERREY

Objetivo









Desarrollo/Metodología



CPU

- DETECCIÓN POR COLOR
- DETECCIÓN DE "OBJETO" MEDIANTE CLASIFICADOR
- VISIÓN ESTÉREO
- LOCALIZACIÓN
- NAVEGACIÓN

MCU

- MONITOREO DE BATERÍA
- Motores
- SERVIDOR UDP
- ALGORITMO MAHONY

USUARIO

- INTERFAZ GRÁFICA

MCU



HiBot TITech M4 Controller

MONITOREO
DE BATERÍA

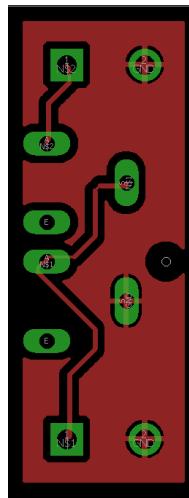
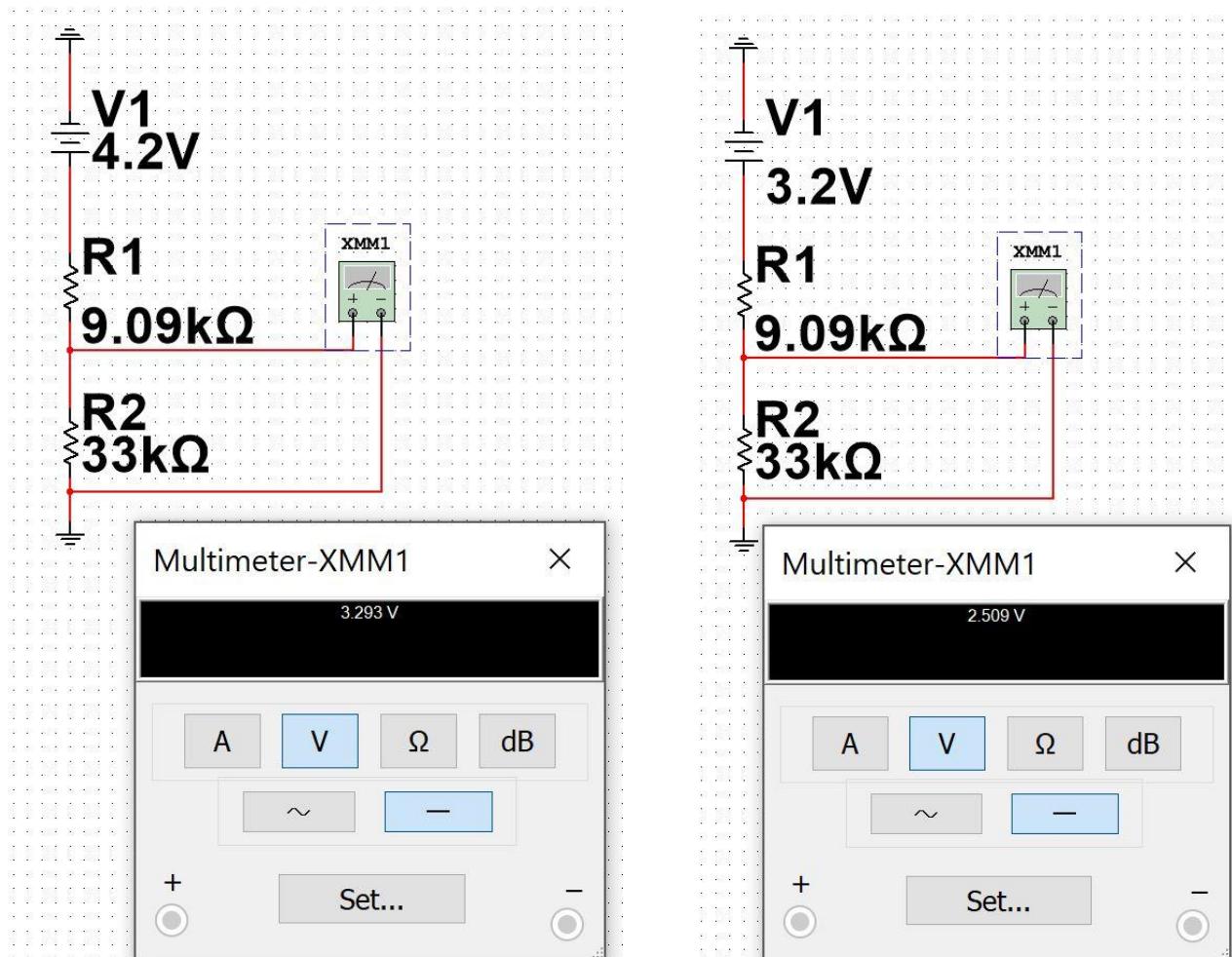
MOTORES

SERVIDOR UDP

ALGORITMO
MAHONY

HiBot, TITech M4 Controller. <http://www.hibot.co.jp/ecommerce/product/detail/8>

MONITOREO DE BATERÍA



```

float measureBattery(){
    float ratio=1.65;
    float pinVoltage=ADCValue[0]*0.0008056640625;
    float batteryVoltage=pinVoltage*ratio;
    return batteryVoltage;
}

float batteryPercentage(){
    float batteryPercentageVal;
    float batteryVoltage=measureBattery();
    float effectiveVoltage=1.2;
    float batteryPercentage=(batteryVoltage-3.2)*100.0/1.2;
    if(batteryPercentage<0){
        batteryPercentageVal=0;
    }
    else if(batteryPercentage>100){
        batteryPercentageVal=100;
    }
    else{
        batteryPercentageVal=batteryPercentage;
    }
    return batteryPercentageVal;
}
    
```

MONITOREO
DE BATERÍA

MOTORES

SERVIDOR UDP

ALGORITMO
MAHONY

MOTORES



```
GPIO_InitTypeDef GPIO_InitStruct ;  
/* Clock for GPIOD */  
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD ,ENABLE) ;  
/* Alternating functions for pins */  
GPIO_PinAFConfig(GPIOD , GPIO_PinSource12 ,GPIO_AF_TIM4) ;  
/* Set pins */  
GPIO_InitStruct .GPIO_Pin=GPIO_Pin_12 ;  
GPIO_InitStruct .GPIO_OType=GPIO_OType_PP ;  
GPIO_InitStruct .GPIO_PuPd=GPIO_PuPd_NOPULL ;  
GPIO_InitStruct .GPIO_Mode=GPIO_Mode_AF ;  
GPIO_InitStruct .GPIO_Speed=GPIO_Speed_100MHz ;  
GPIO_Init(GPIOD,&GPIO_InitStruct ) ;
```

Configuración pin para PWM

MOTORES

```
void PWM_TIMER_Init(void){
    TIM_TimeBaseInitTypeDef TIM_BaseStruct;

    /*Enable clock for TIM4*/
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4,ENABLE);

    TIM_BaseStruct.TIM_Prescaler=0;
    /*Count up*/
    TIM_BaseStruct.TIM_CounterMode=TIM_CounterMode_Up;

    TIM_BaseStruct.TIM_Period=8399;
    TIM_BaseStruct.TIM_ClockDivision=TIM_CKD_DIV1;
    TIM_BaseStruct.TIM_RepetitionCounter=0;
    /*Initialize TIM4*/
    TIM_TimeBaseInit(TIM4,&TIM_BaseStruct);
    /*Start count on TIM4*/
    TIM_Cmd(TIM4,ENABLE);
}
```

Configuración Timer para PWM

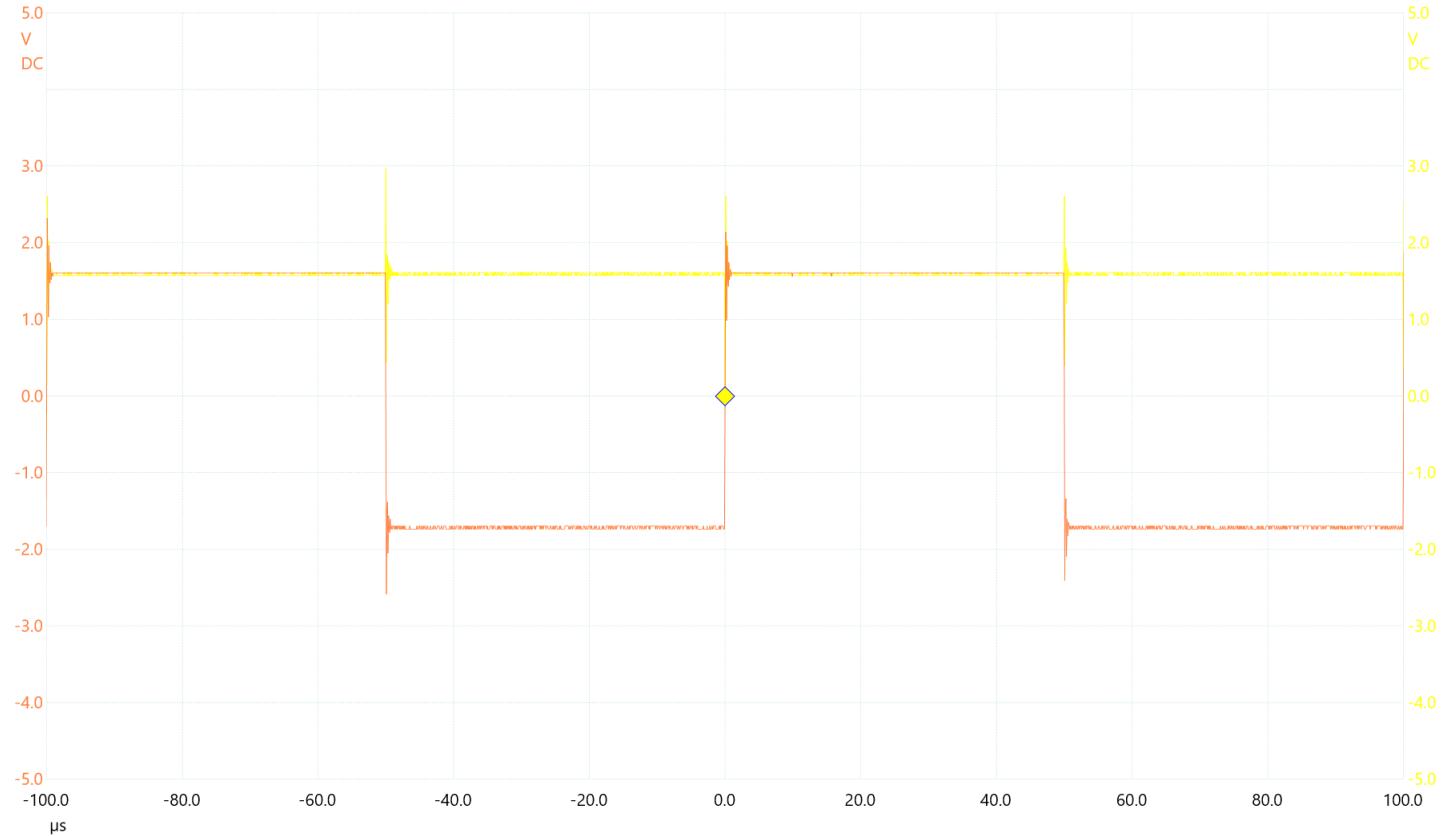
```
void PWM_Init1(int Tim_PeriodCH1){
    TIM_OCInitTypeDef TIM_OCStruct;
    /*Common settings*/
    /*PWM mode 2 = Clear on compare match*/
    /*PWM mode 1 = Set on compare match*/
    TIM_OCStruct.TIM_OCMode=TIM_OCMode_PWM2;
    TIM_OCStruct.TIM_OutputState=TIM_OutputState_Enable;
    TIM_OCStruct.TIM_OCPolarity=TIM_OCPolarity_Low;
    /*
     * To get proper duty cycle , you have simple equation
     *
     * pulse_length=((TIM_Period+1)*DutyCycle)/100-1
     *
     * where DutyCycle is in percent , between 0 and 100%
     *
     * 25% duty cycle:2099
     * 50% duty cycle:4199
     * 75% duty cycle:6299
     * 100% duty cycle:8399
     */
    TIM_OCStruct.TIM_Pulse=Tim_PeriodCH1;
    TIM_OC1Init(TIM4,&TIM_OCStruct);
    TIM_OC1PreloadConfig(TIM4,TIM_OCPreload_Enable);
}
```

Asignación de PWM


PD14(50%) PD15(100%)

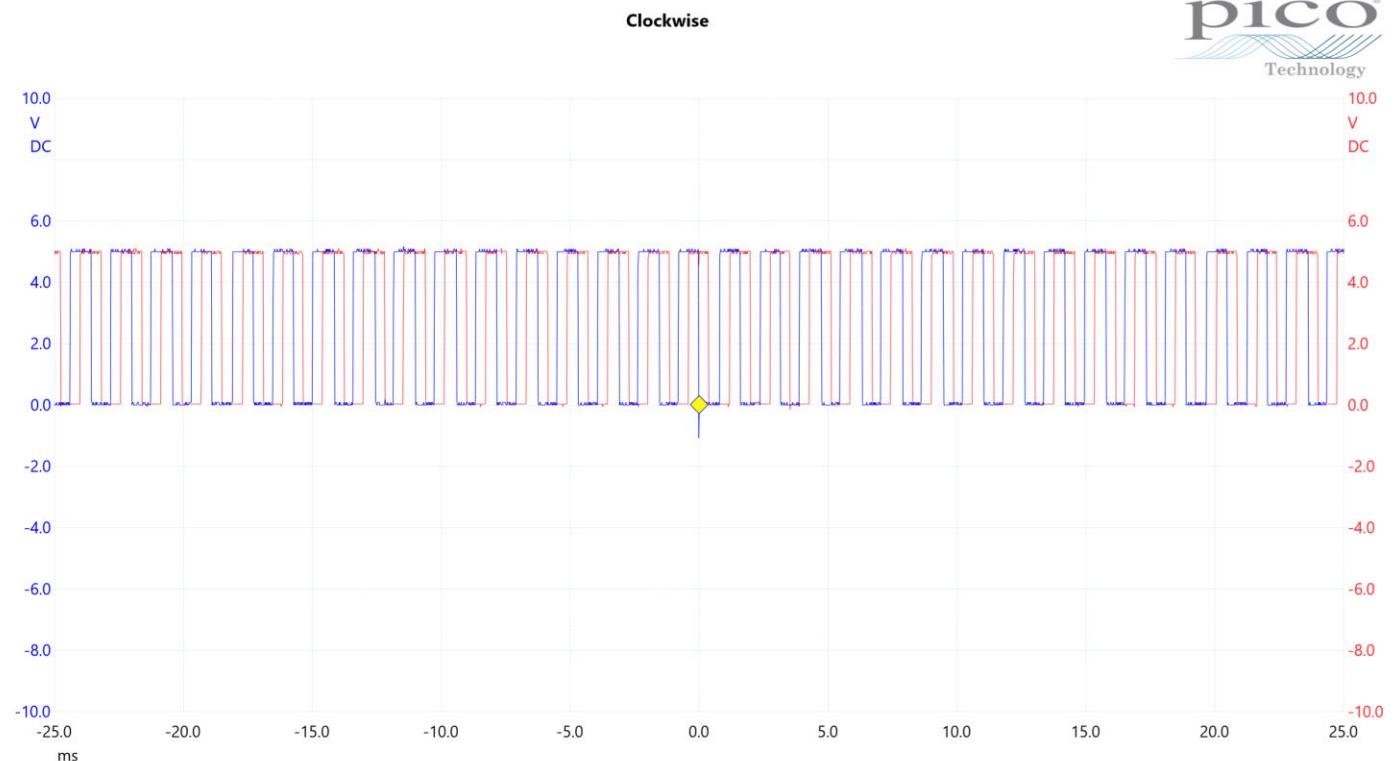
Naranja: Channel A, PD14

Amarillo: Channel B, PD15



Channel	Name	Span	Value	Min	Max	Average	σ
A	Duty Cycle	Whole trace	50.08 %	50.08 %	50.08 %	50.08 %	0 %
B	Duty Cycle	Whole trace	100 %	100 %	100 %	100 %	0 %
A	Frequency	Whole trace	10 kHz	10 kHz	10 kHz	10 kHz	0 Hz

MOTORES



Blue->Channel A
Red->Channel B

MONITOREO
DE BATERÍA

MOTORES

SERVIDOR UDP

ALGORITMO
MAHONY

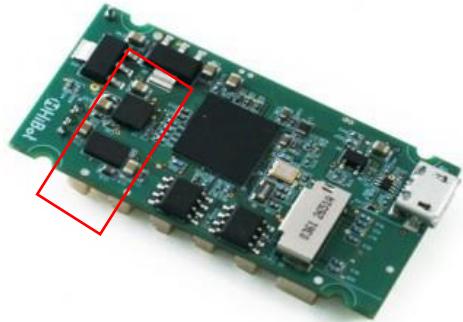
ALGORITMO MAHONY

Filtro complementario que requiere sólo salidas de acelerómetro y giroscopio, apropiado para su uso en hardware embebido, estima la postura y orientación así como también el sesgo del giroscopio online.

Salidas:

Cuaterniones

Roll Pitch Yaw



ALGORITMO
MAHONY

ACELERÓMETRO

LSM303DLHC

IMU

GIROSCOPIO

L3G4200D

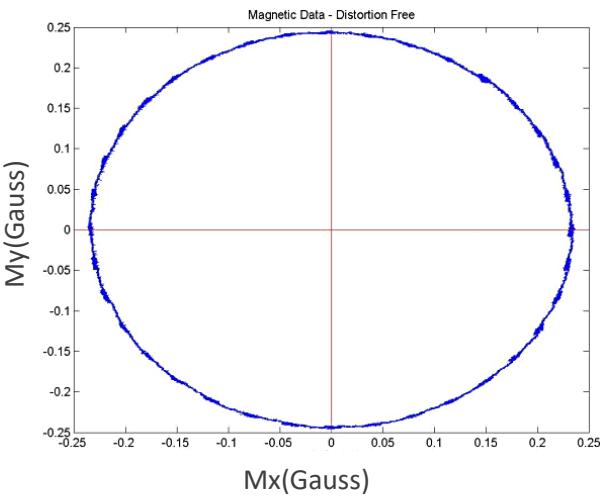
MAGNETÓMETRO

LSM303DLHC

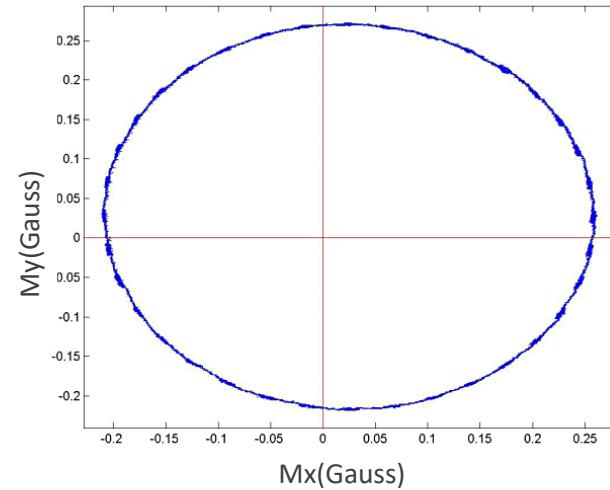
HiBot, TITech M4 Controller. <http://www.hibot.co.jp/ecommerce/product/8>

ALGORITMO MAHONY

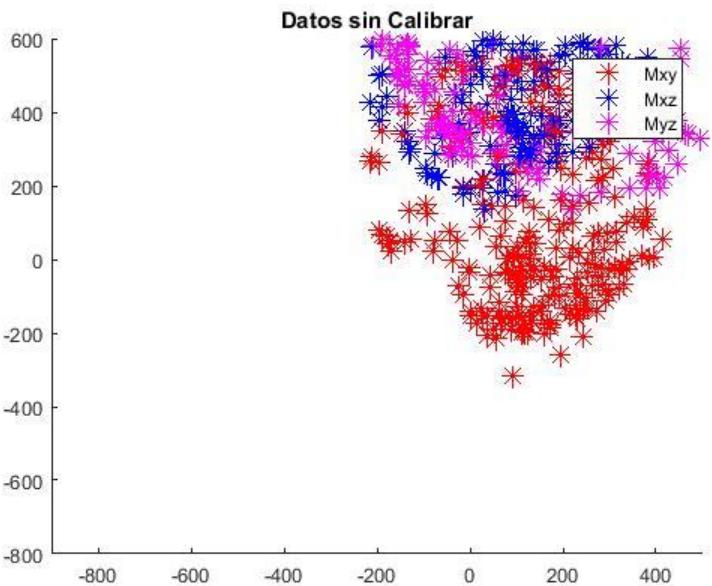
MAGNETÓMETRO



Lectura ideal de
magnetómetro

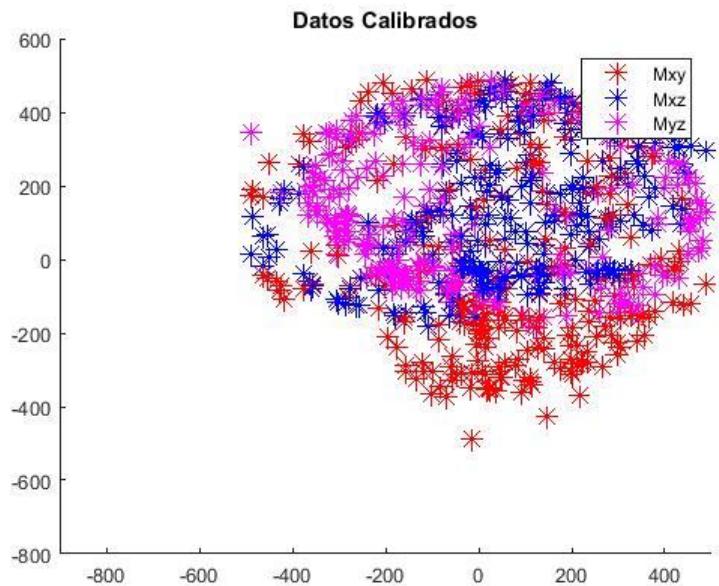


Distorsión hard-iron



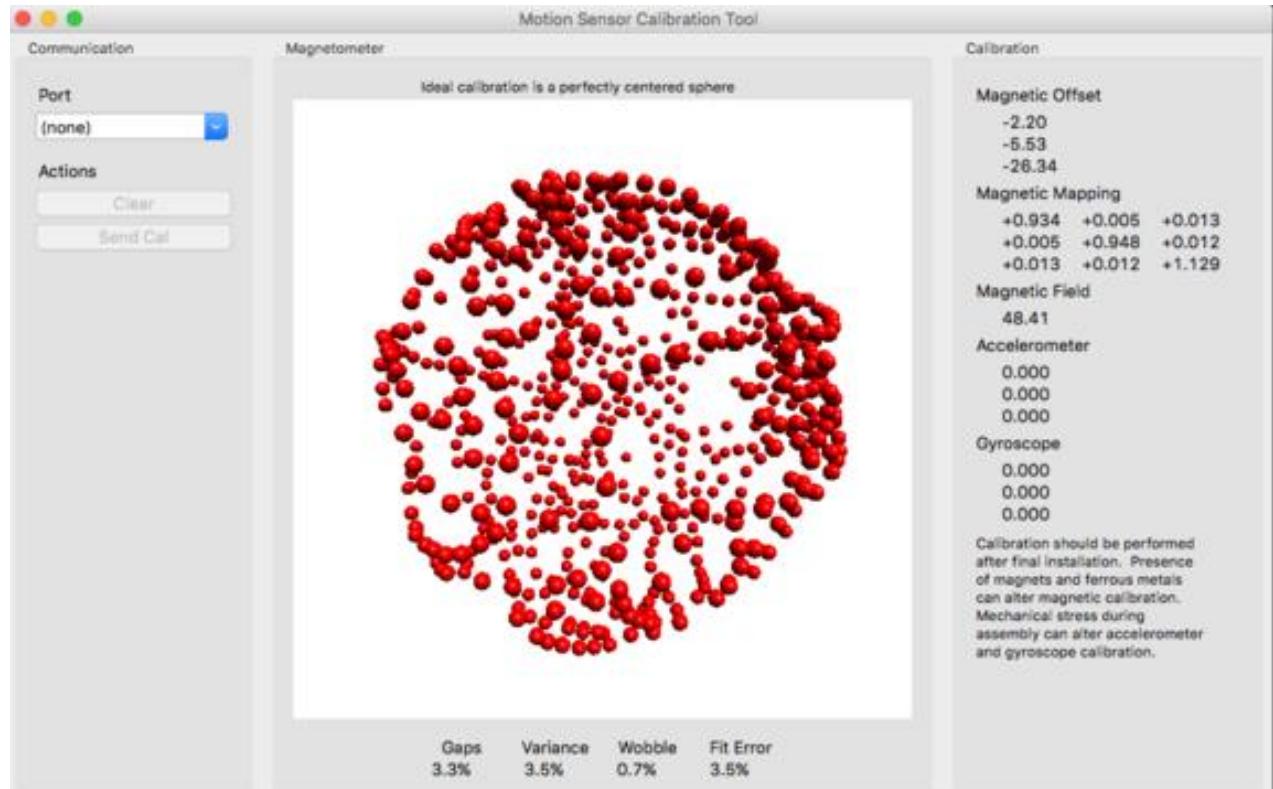
$$\alpha = \frac{(x_{max} + x_{min})}{2}$$

$$\beta = \frac{(y_{max} + y_{min})}{2}$$



Mxy
Mxz
Myz

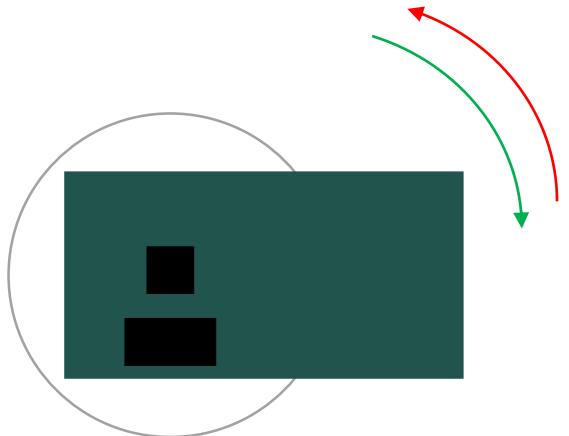
$x = \text{mG (milliGauss)}$
 $y = \text{mG}$



~~MAGNETOMETRO~~

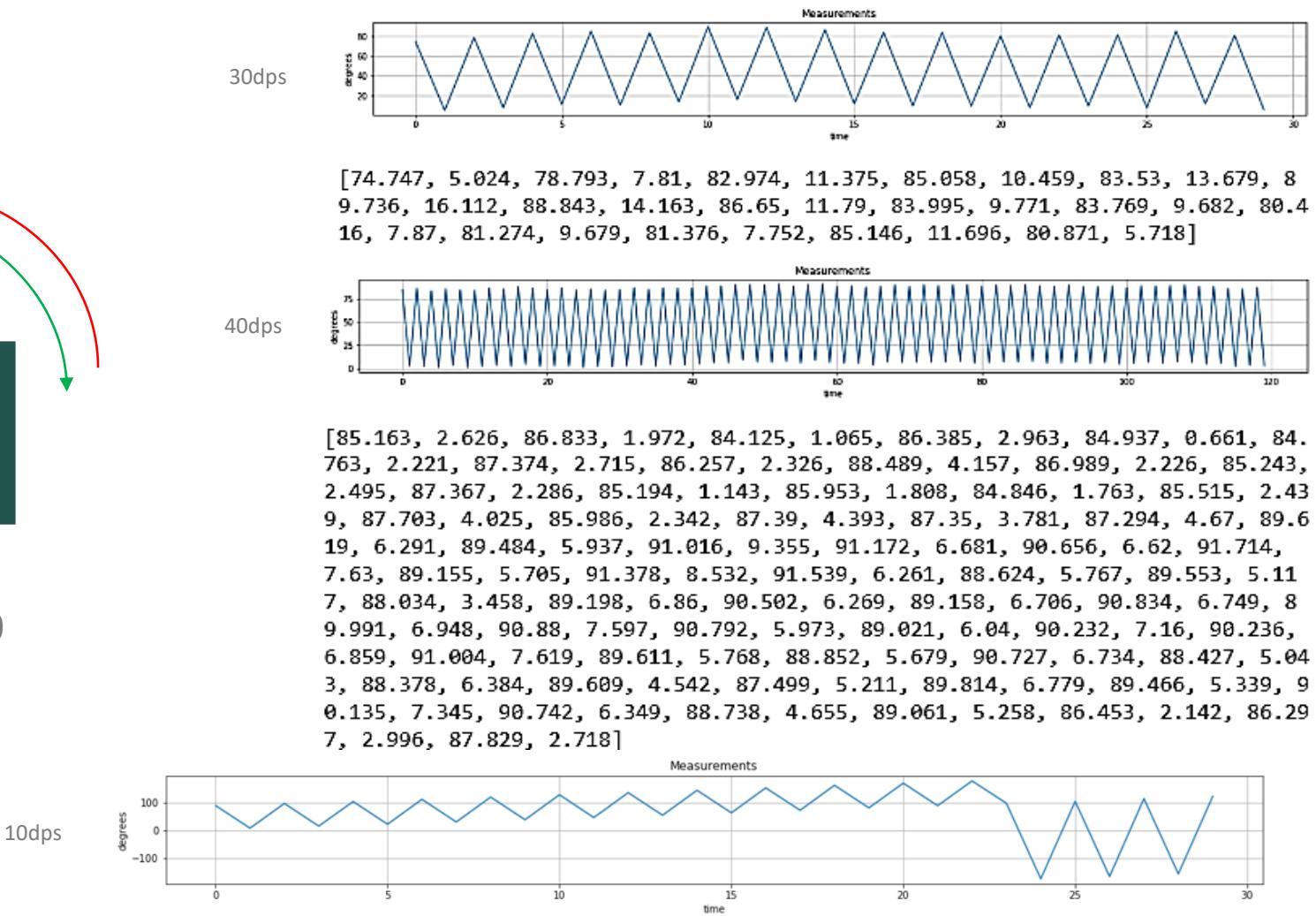
Townsend, K. (2016). AHRS for Adafruit's 9-DOF, 10-DOF, or LSM9DS0 breakouts into a low cost AHRS

**ALGORITMO
MAHONY**



Iteraciones de 0-90
grados

x = número de iteración
 y = ángulo yaw obtenido del algoritmo
dps=degrees per second

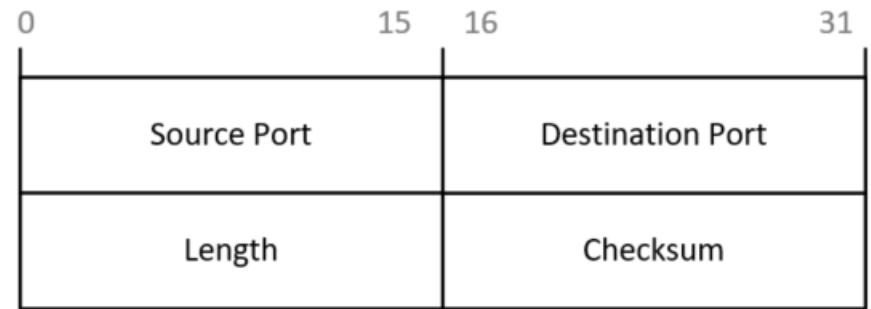
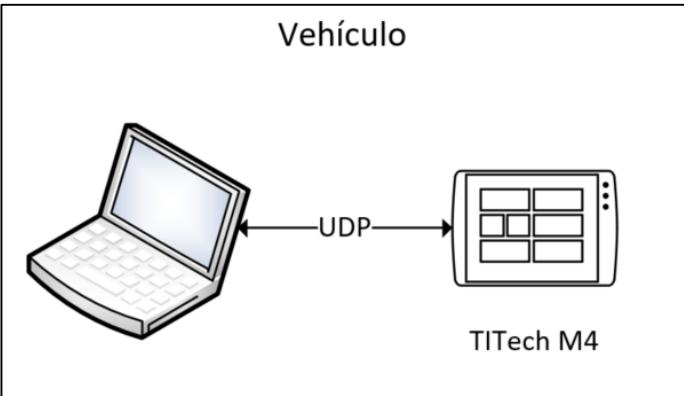


MONITOREO
DE BATERÍA

MOTORES

SERVIDOR UDP

ALGORITMO
MAHONY



SERVIDOR UDP

```

//-----UDP data send-----
char UDParray[32];
sprintf(UDParray,"%3.0f,%3.0f,%3.0f,%3.0f,%i,%i,",getYaw(),getRoll(),getPitch(),batteryPercentage(),encoderARead(),encoderBRead());
udp_server_data_send(UDParray,32);

try:
    udp,addr=udpSock.recvfrom(1024)

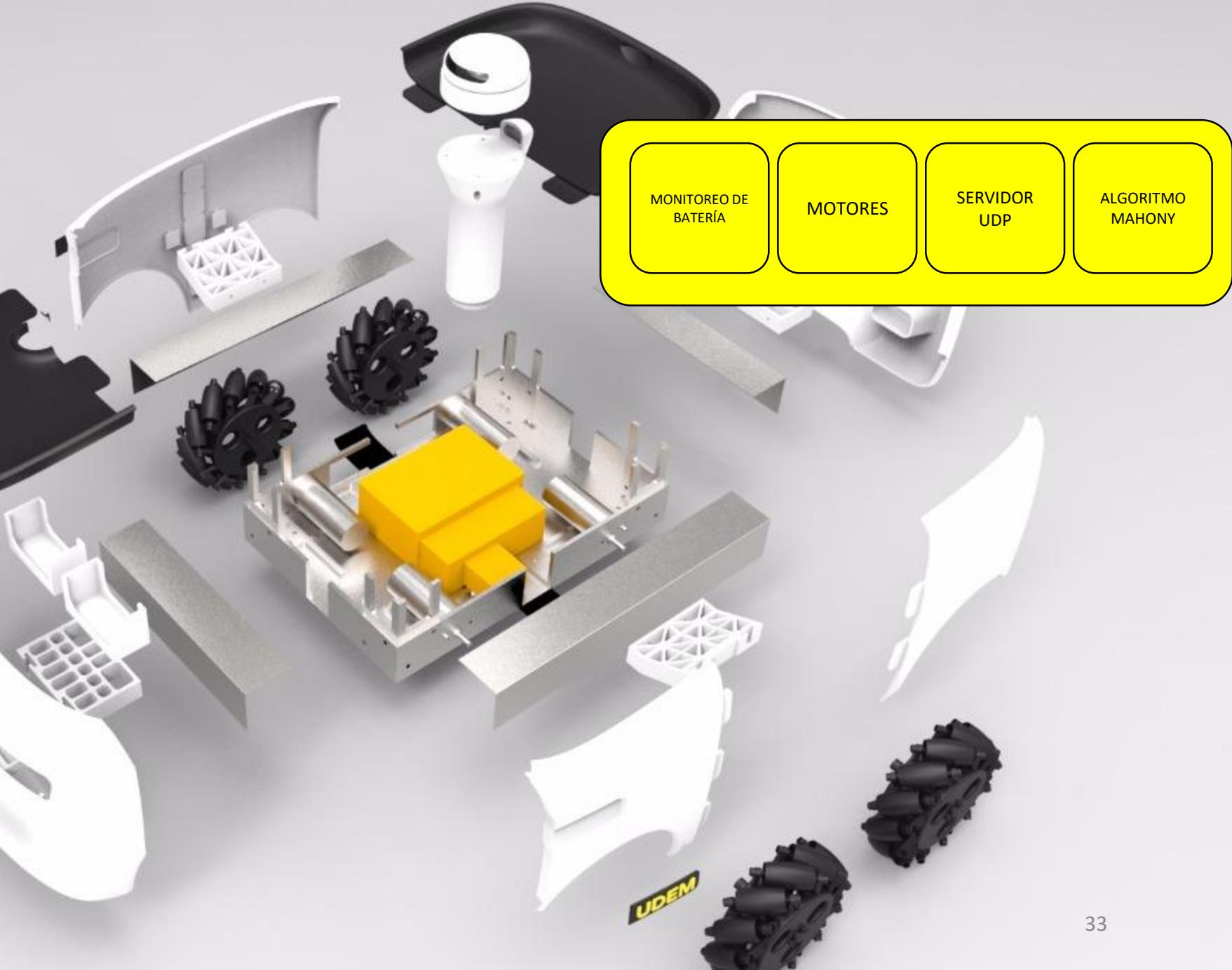
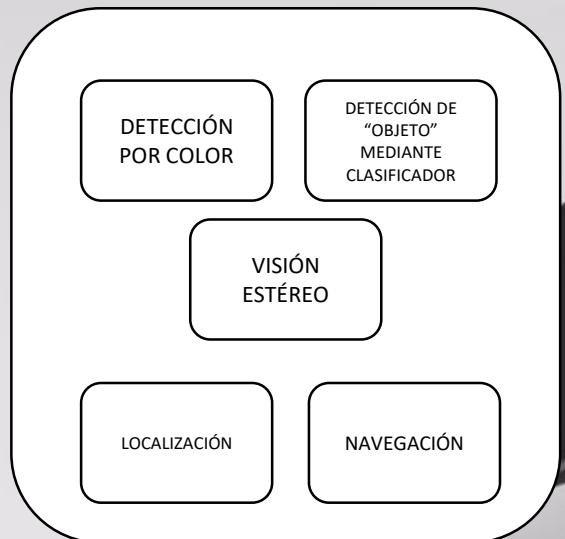
    udp=udp.split(",") #UDP data is splitted
    yaw=str(udp[0])
    roll=str(udp[1])
    pitch=str(udp[2])
  
```

MONITOREO
DE BATERÍA

MOTORES

SERVIDOR UDP

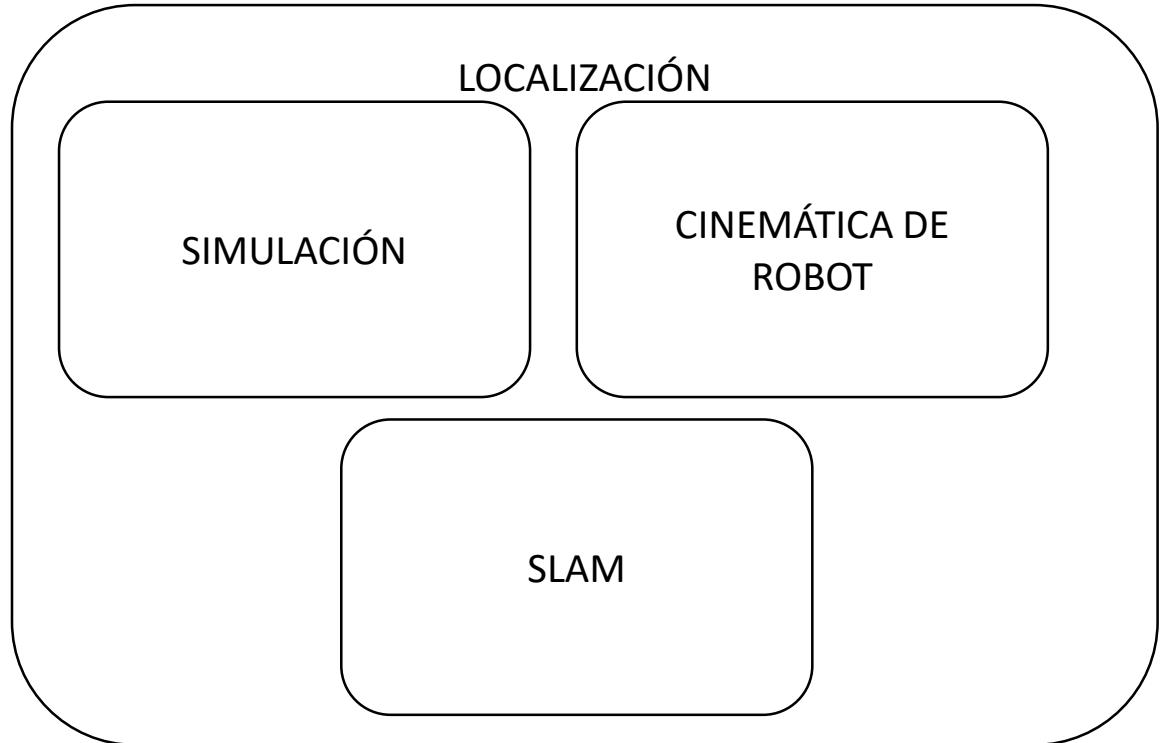
ALGORITMO
MAHONY



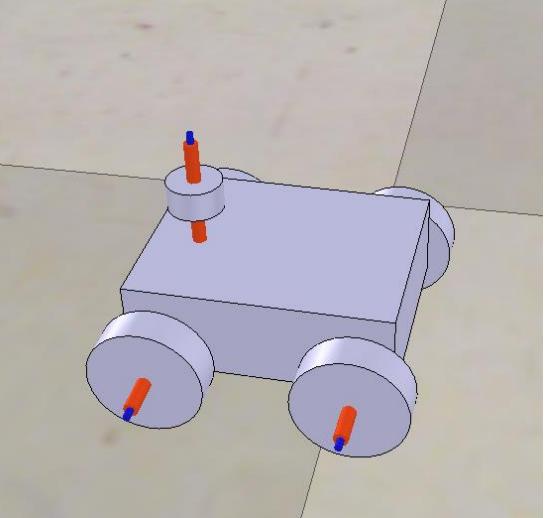
Unidad central de procesamiento

LOCALIZACIÓN

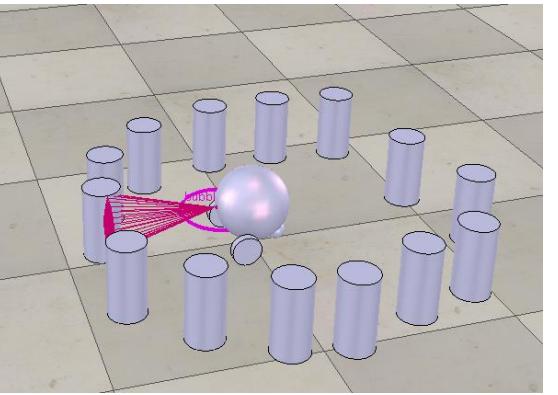
NAVEGACIÓN



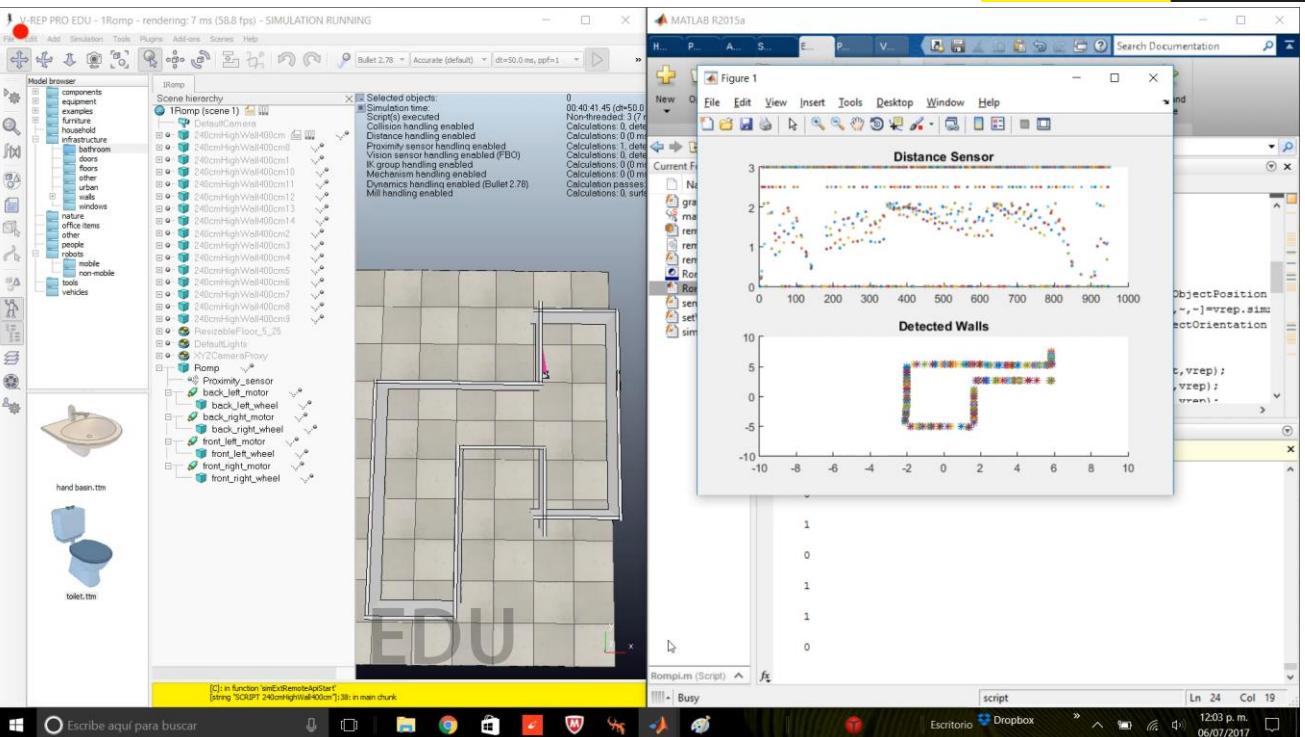
SIMULACIÓN



Modelo de un robot diferencial



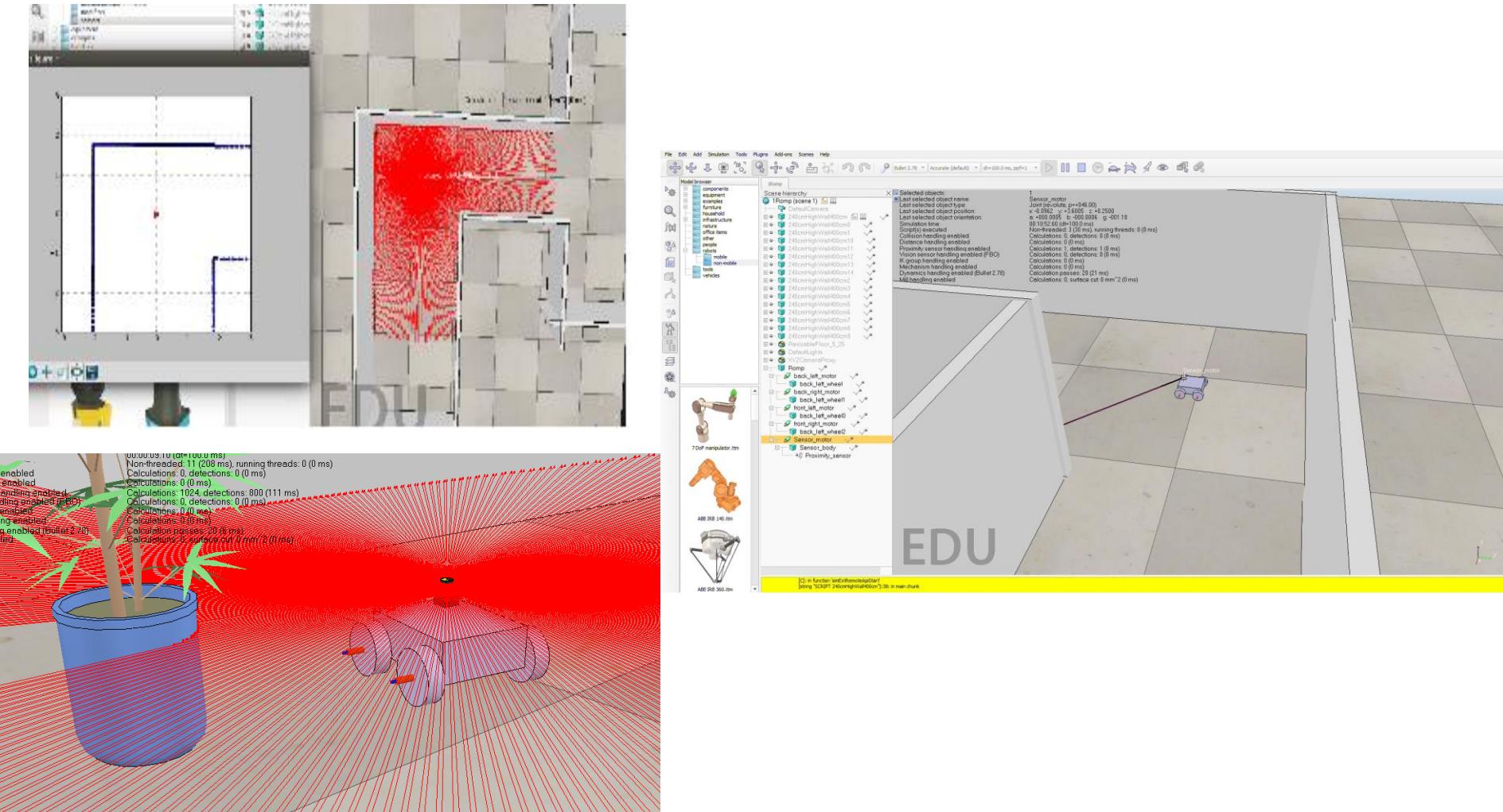
Simulación con figuras primitivas

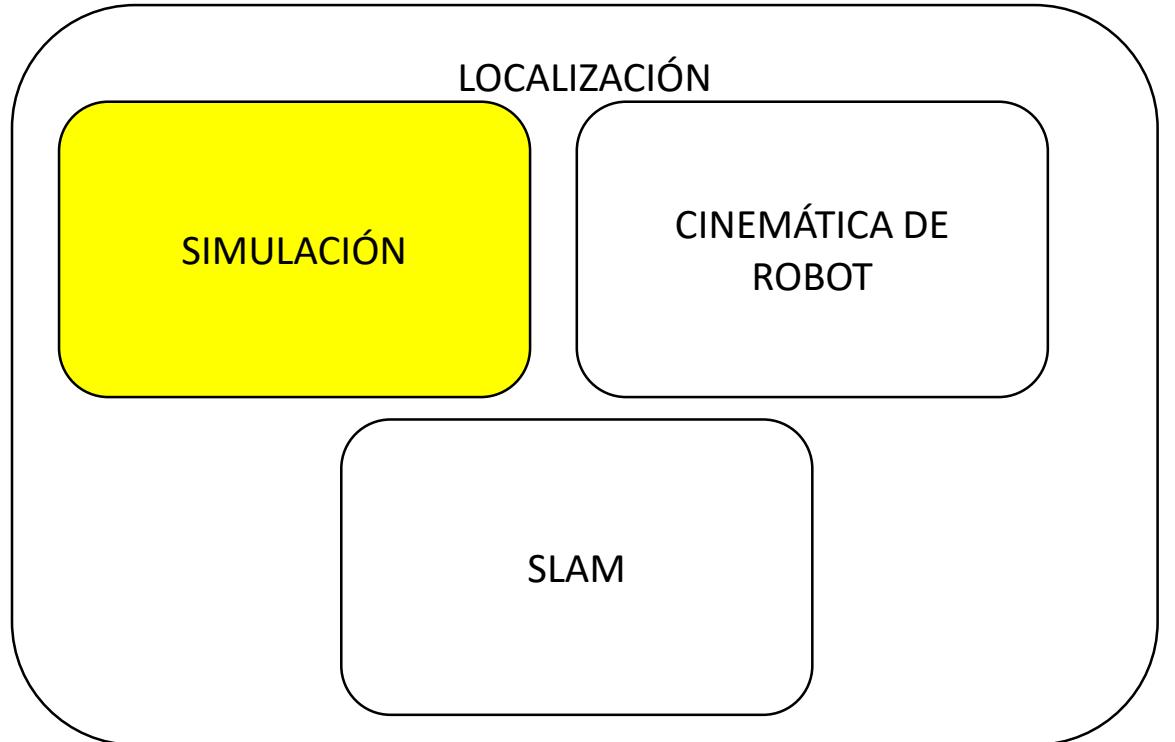


$$\hat{y}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} r\cos(\theta + \theta_v) + x_v \\ r\sin(\theta + \theta_v) + y_v \end{bmatrix}.$$

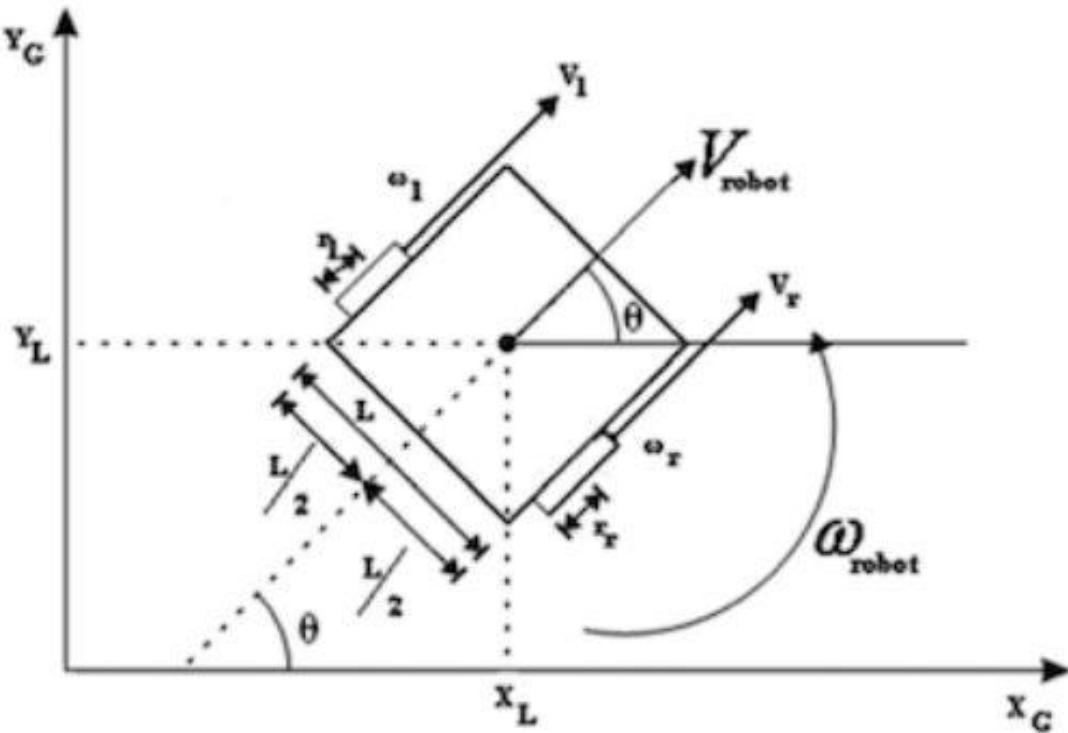
Francisco, M. A. R., & Antoni, G. S. (2013). SLAM con mediciones angulares: método por triangulación estocástica. *Ingeniería, investigación y tecnología*, 14(2), 257-274.

SIMULACIÓN





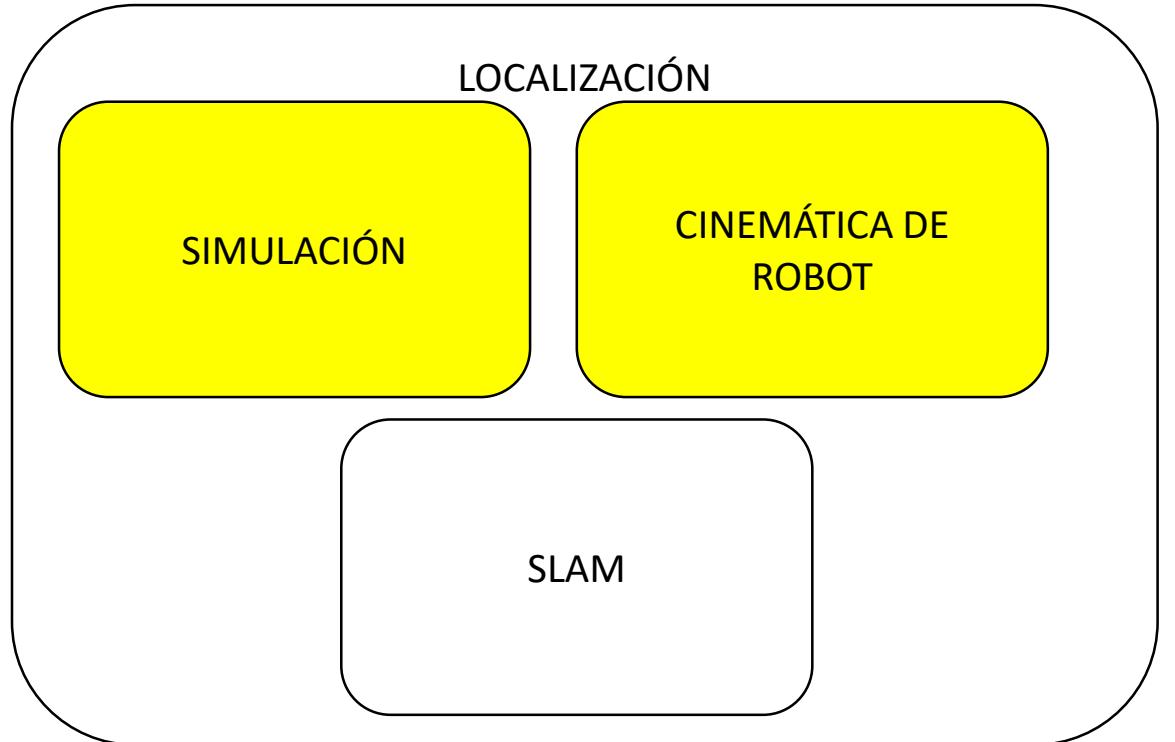
CINEMÁTICA DE
ROBOT



$$\dot{x} = \frac{R}{2}(v_r + v_l)\cos\theta$$

$$\dot{y} = \frac{R}{2}(v_r + v_l)\sin\theta$$

$$\dot{\phi} = \frac{R}{L}(v_r - v_l)$$



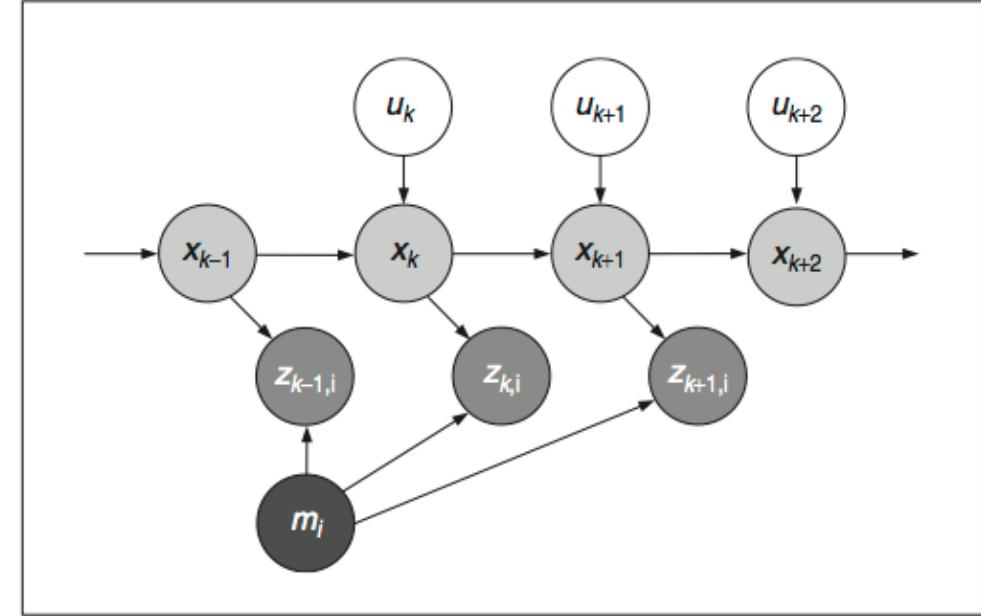
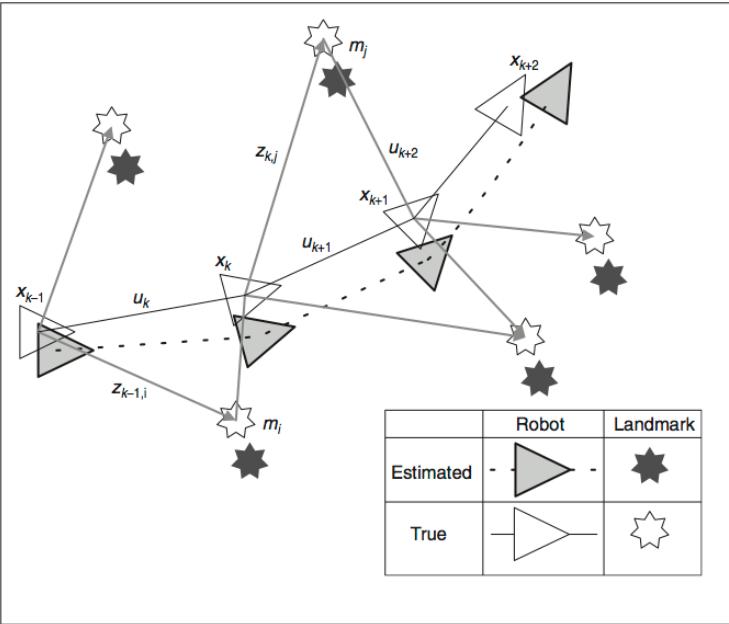
Estimación de la posición

Estimación de la
posición

$$\begin{bmatrix} x_{v(k+1)} \\ y_{v(k+1)} \\ \theta_{v(k+1)} \end{bmatrix} = \begin{bmatrix} x_{v(k)} + \partial \cos(\theta') \\ y_{v(k)} + \partial \sin(\theta') \\ \Theta_{v(k)} + \theta' \end{bmatrix}$$

- Ruido en los sensores
- Desplazamiento impreciso del robot
- Simetrías en el entorno
- Observabilidad parcial
- Ambiente dinámico.

Algoritmo SLAM



Raspberry Pi GoPiGo Robot EKF SLAM. Obtenido en línea de:
<http://gopalmenon.github.io/Raspberry-Pi-GoPiGo-Robot-EKF-SLAM-Manuscript/>

Estimación de la posición

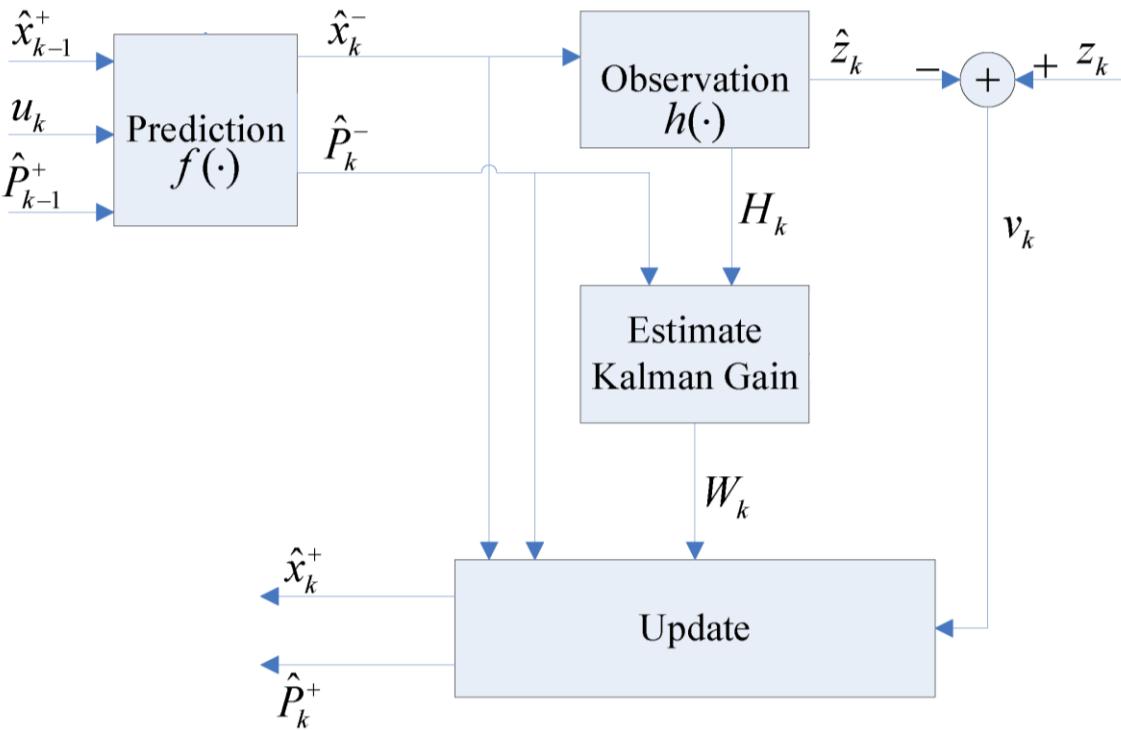
Estimación de la
posición

$$h_i(\hat{x}_v, \hat{y}_i) = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_v)^2 + (y_i - y_v)^2} \\ atan2(y_i - y_v, x_i - x_v) - \theta_v \end{bmatrix} + \begin{bmatrix} w_r \\ w_\beta \end{bmatrix}$$

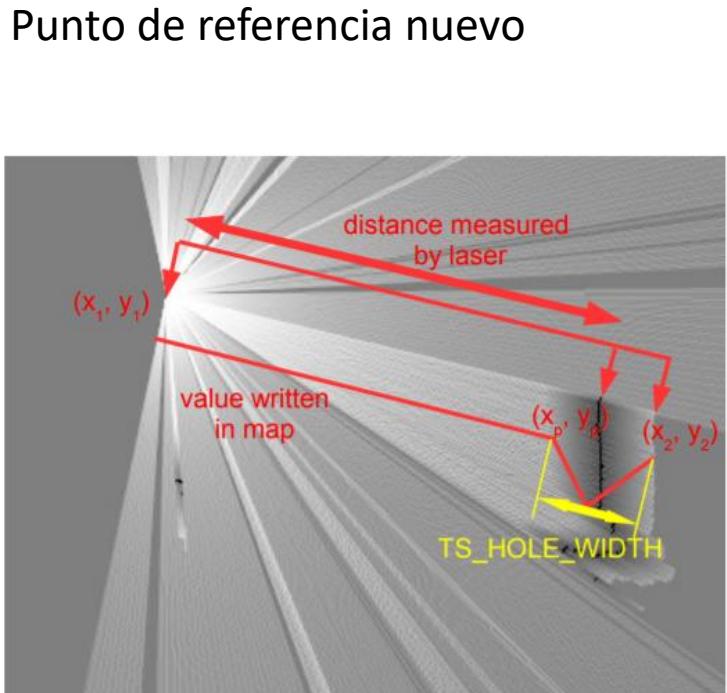
Dónde $\omega = (w_r, w_\beta)^T$ es una variable aleatoria de distribución Gaussiana con media cero

Francisco, M. A. R., & Antoni, G. S. (2013). SLAM con mediciones angulares: método por triangulación estocástica. *Ingeniería, investigación y tecnología*, 14(2), 257-274.

Algoritmo SLAM



Core SLAM



Algoritmo de Monte Carlo

$$x_i^+ = f(x_{i(a)}, u_{\langle k \rangle} + q_{\langle k \rangle})$$

$$v_i = z_{\langle k+1 \rangle} - h(\hat{x}_i^+, p_j)$$

$$w_i = e^{-v_i^{T_L-1}} + w_0$$

$$w'_i = w_i / \sum_{i=1}^N w_i$$

$$x_{\langle k+1 \rangle} \rightarrow R(x_{\langle k+1 \rangle}^+, \omega)$$

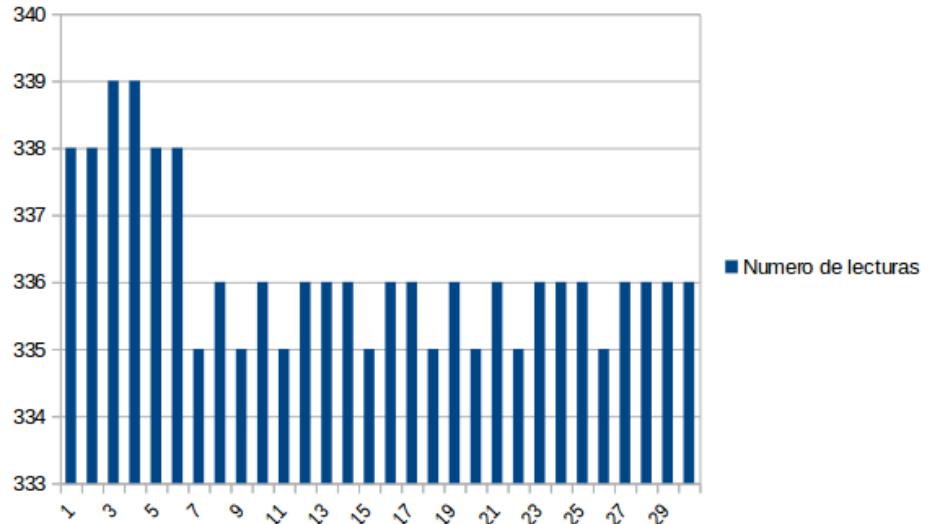
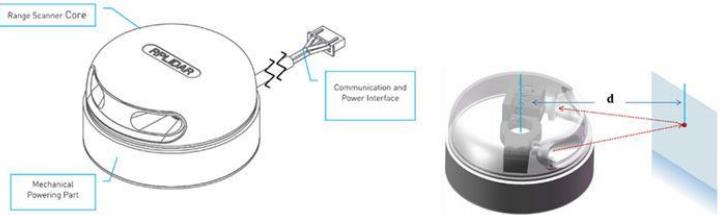
1. Posición estimada + número aleatorio
2. Determinar diferencia entre lectura y lectura estimada
3. Estimar que tan probable es que la lectura sea correcta
4. Normalizar todas las lecturas
5. Hacer un re-muestreo de las posiciones estimadas

SLAM

Modificación de
Breezy SLAM

- Crear una clase correspondiente al tamaño de las llantas, separación y numero de pulsos por vuelto completo del encóder del robot UDEM.
- Algoritmo para comunicación entre Script C para leer encoders y versión Python de BreezySLAM.
- Aplicación para enviar datos desde Script C Sharp a Python
- Script de Python para leer sensores desde simulación

RPLIDAR



LOCALIZACIÓN

SIMULACIÓN

CINEMÁTICA DE
ROBOTS

SLAM

LOCALIZACIÓN

NAVEGACIÓN

NAVEGACIÓN

MÉTODOS
PROBABILISTICOS

RRT

MÉTODOS PROBABILISTICOS

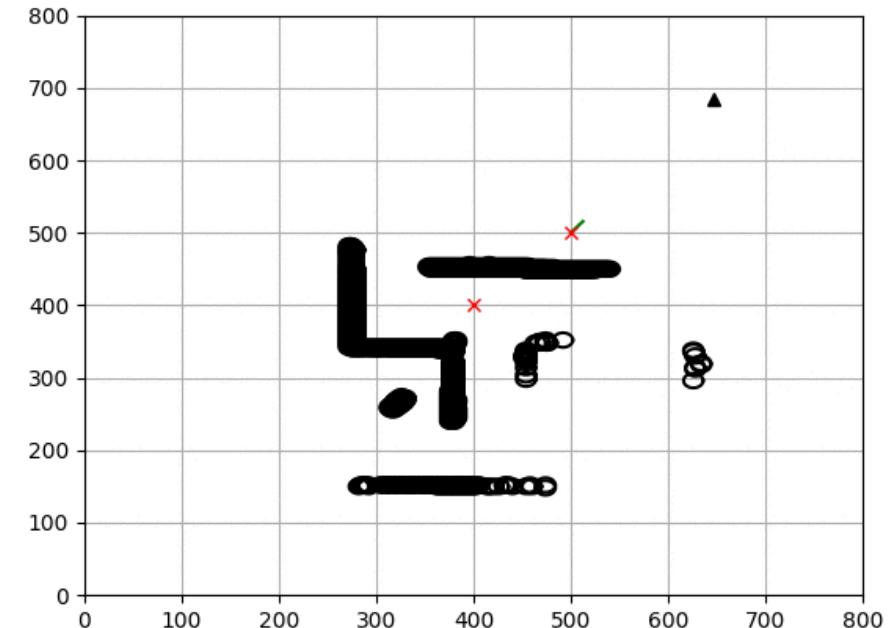
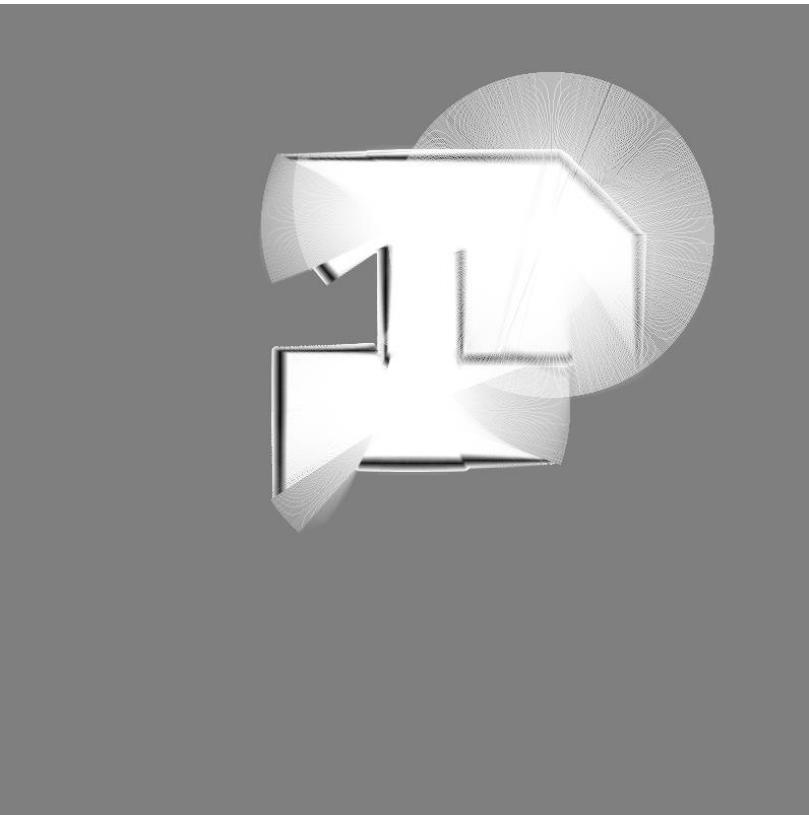
NAVEGACIÓN

MÉTODOS
PROBABILISTICOS

RRT

Rapidly Expanding Random Trees

RRT



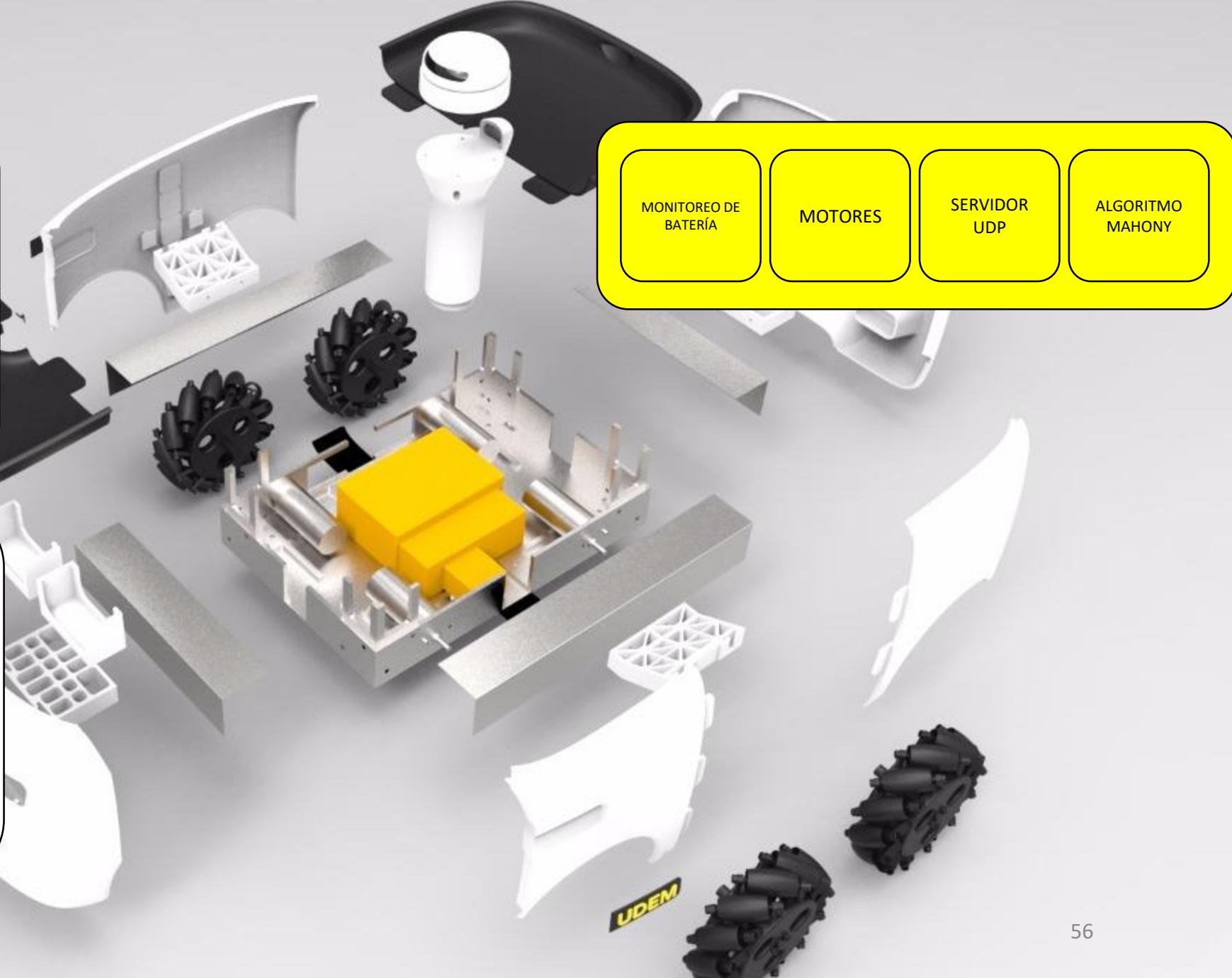
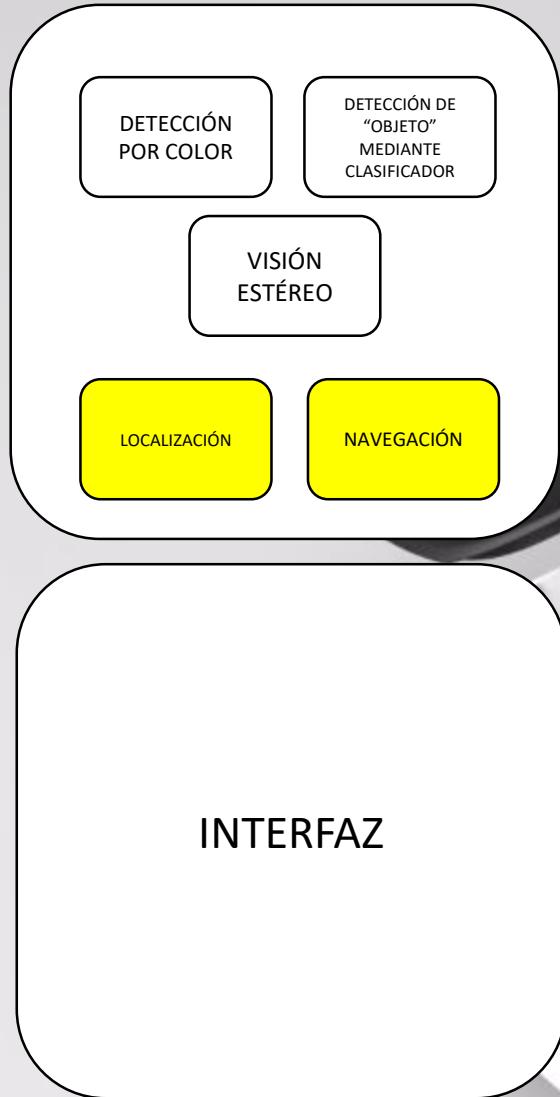
NAVEGACIÓN

MÉTODOS
PROBABILISTICOS

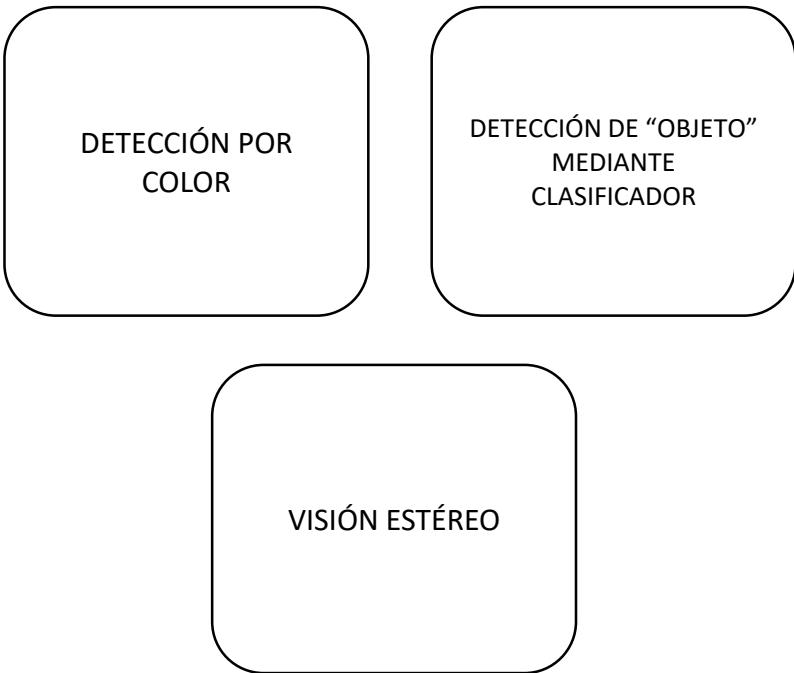
RRT

LOCALIZACIÓN

NAVEGACIÓN



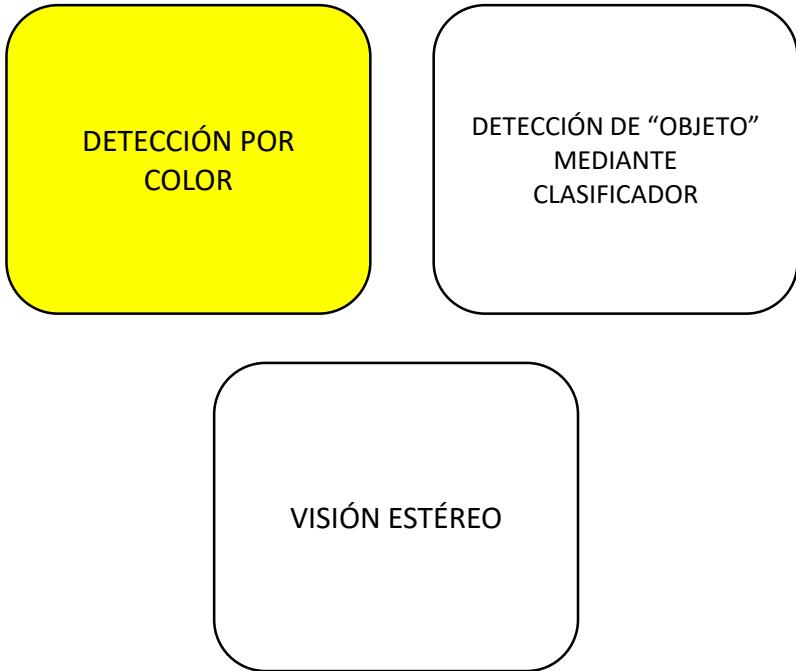
Sistema de visión



Cambiar el espacio de color para sementar la imagen.

1. Cambiar el espacio de color RGB → HSV.
 2. Modificar los valores de las matrices HSV.
mapa binario que contenga el objeto a
 3. Aplicar la máscara binaria.
 4. Extraer el objeto en el
 5. Obtener el mapa
 6. Determinar el color.
 7. Realizar alguna acción sobre la imagen (trazar un rectángulo marcando la detección)
- 

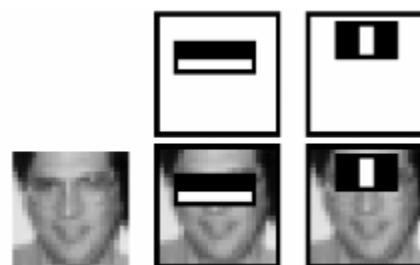
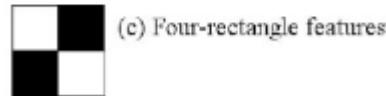
DETECCIÓN POR
COLOR



DETECCIÓN DE “OBJETO”
MEDIANTE
CLASIFICADOR

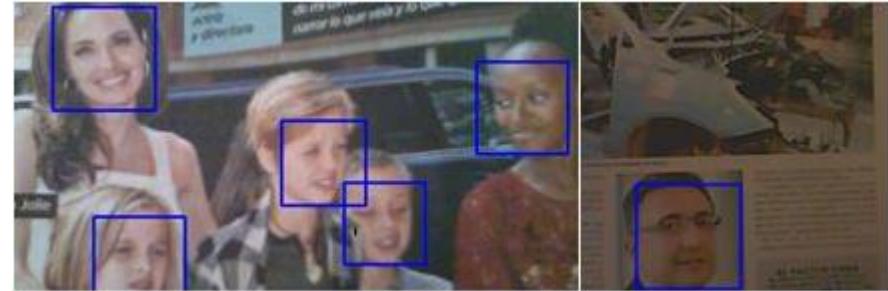
Clasificador Características tipo Haar Cascada

Detección de un “objeto” mediante un clasificador en cascada con características tipo Haar propuesto por Paul Viola y Michael Jones.



Fuente: OpenCV

Detección de rostros



DETECCIÓN DE “OBJETO”
MEDIANTE
CLASIFICADOR

Utilización de un clasificador en cascada con características tipo Haar pre-entrenado desarrollado por Paul Viola y Michael Jones.



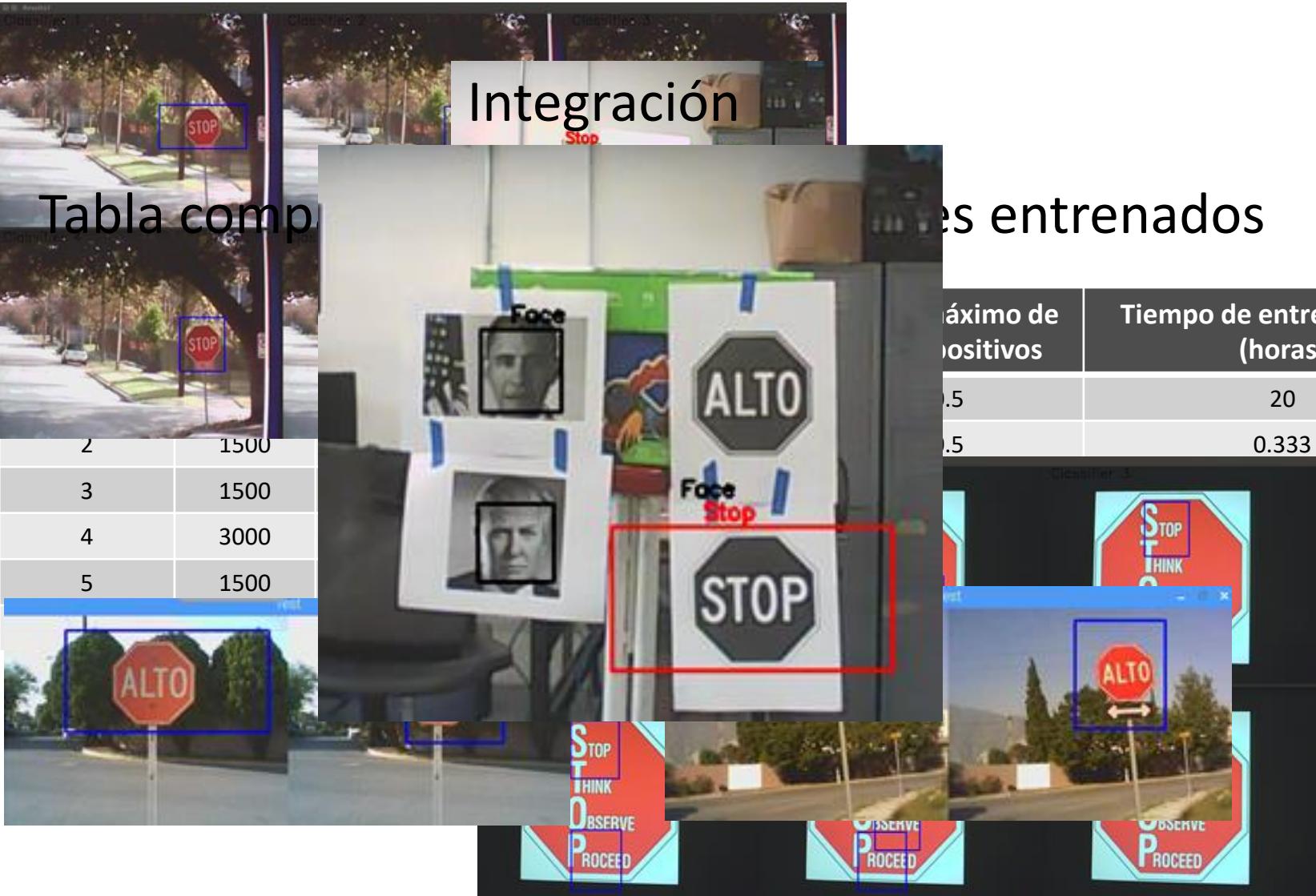


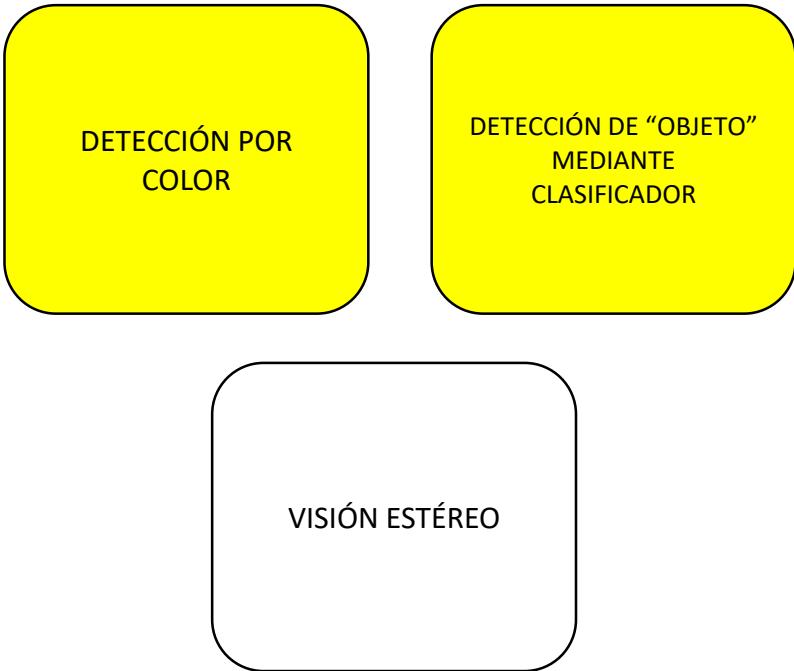
DETECCIÓN DE “OBJETO”
MEDIANTE
CLASIFICADOR

Entrenamiento de un clasificador

1. Cargar imágenes positivas.
2. Cargar imágenes negativas.
3. Crear las muestras de entrenamiento.
4. Juntar las muestras en un sólo archivo.
5. Definir parámetros del entrenamiento.
6. Clasificador entrenado “.xml”.

DETECCIÓN DE "OBJETO"
MEDIANTE
CLASIFICADOR







La visión estéreo utiliza y procesa dos imágenes del mismo lugar, al mismo tiempo, pero tomadas por cámaras diferentes en diferentes posiciones.

Aplicado para estimar profundidad y evadir obstáculos.

1. Calibración
2. Rectificación
3. Mapa de disparidad
4. Estimación de profundidad de un objeto

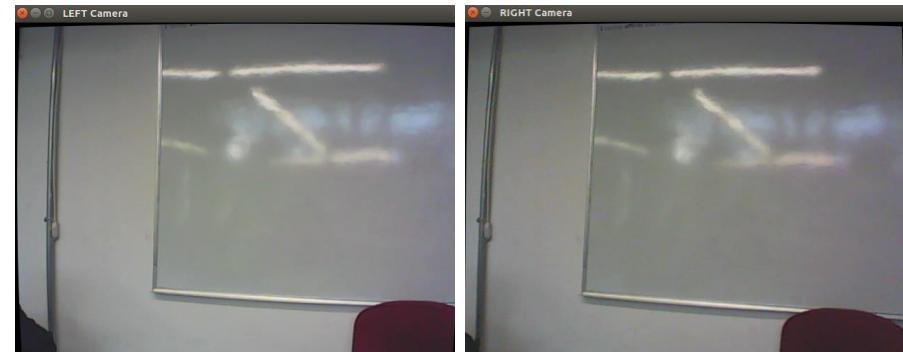


1. Calibración

Calibración de parámetros intrínsecos y extrínsecos.

VISIÓN ESTÉREO

Corrección de la imagen



2. Rectificación

Las imágenes calibradas de las cámaras se posicionan sobre el mismo plano.



3. Mapa de disparidad

Mapa a escala de grises que varía la intensidad dependiendo la distancia del objeto.

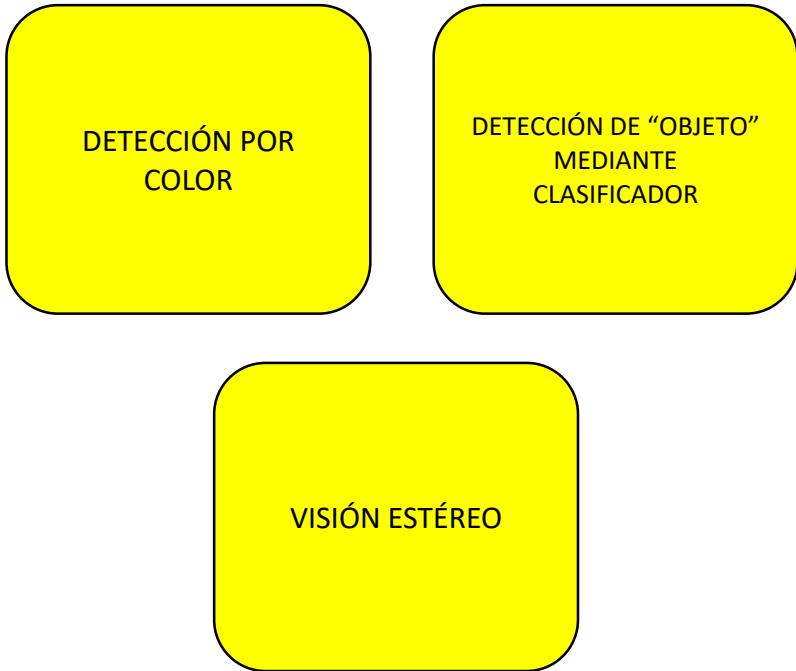


4. Estimación de profundidad de un “objeto”

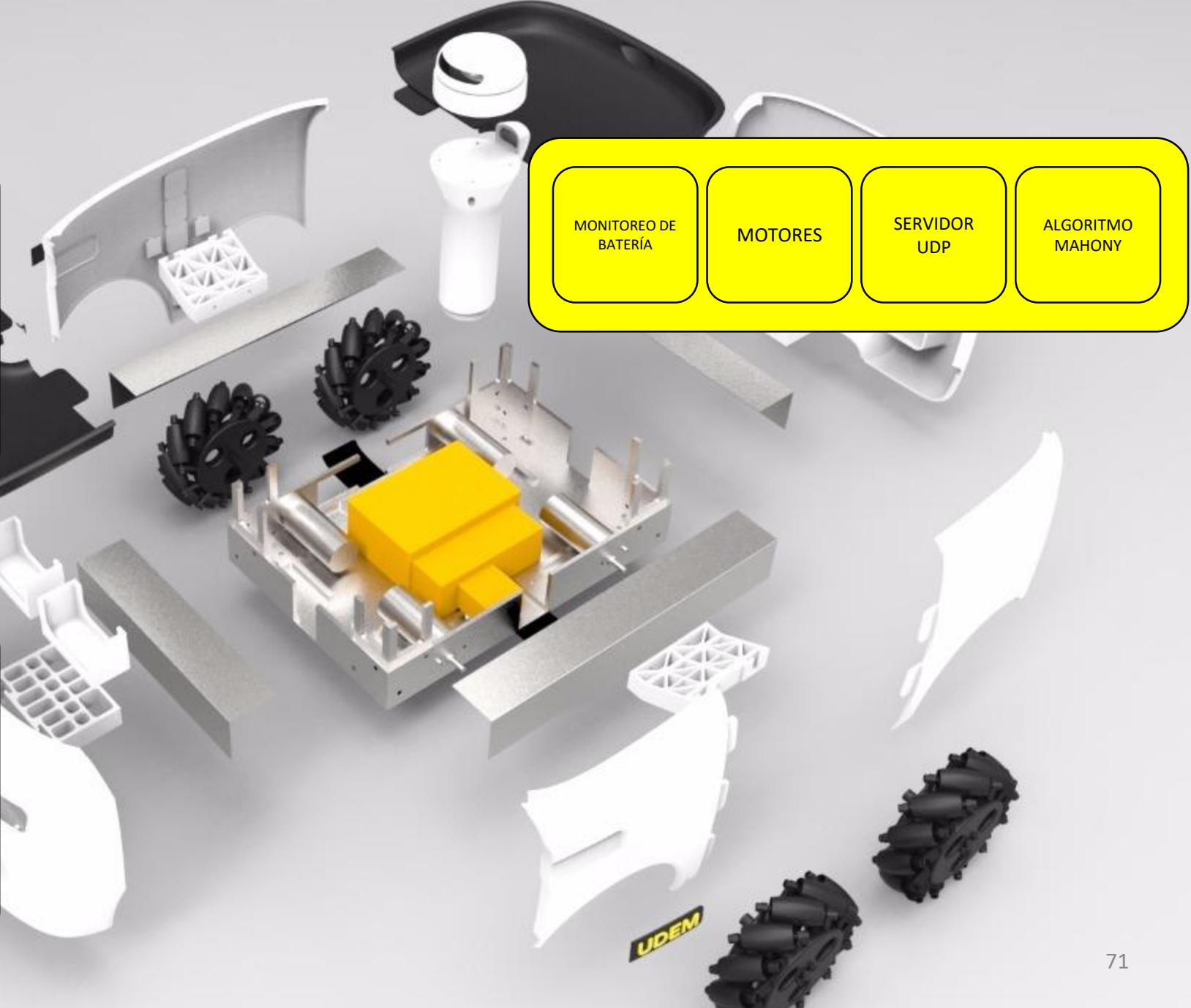
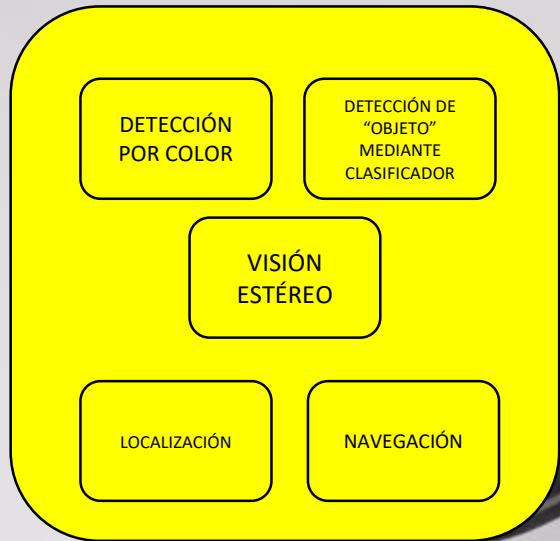
VISIÓN ESTÉREO

Objeto	Distancia real	Distancia estimada
1. Hoja de papel	106 cm	x:106.1 cm ; σ : 0.5043 cm
2. Detergente	79 cm	x: 80.2 cm ; σ : 1.205 cm
3. Cinta métrica	96 cm	x: 94 cm ; σ : 0 cm
4. Bote de pintura	77 cm	x: 79.4 cm ; σ : 1.78 cm
1. Hoja de papel	106 cm	x: 104.4 cm ; σ : 4.16 cm





INTERFAZ

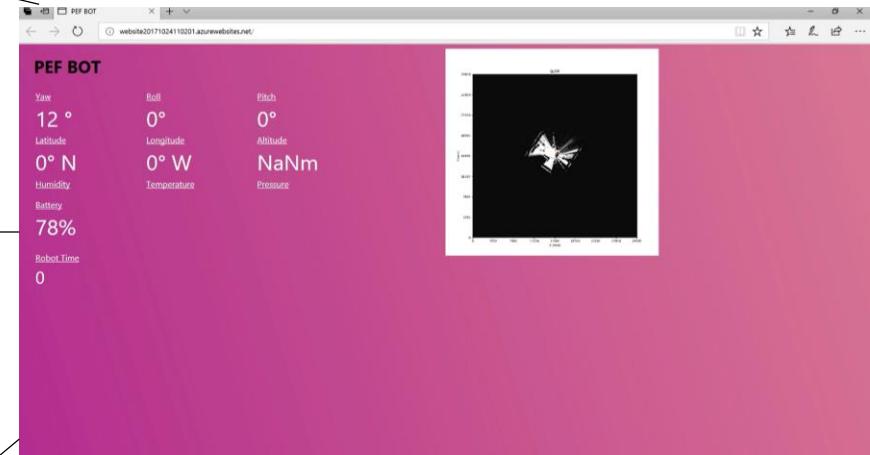


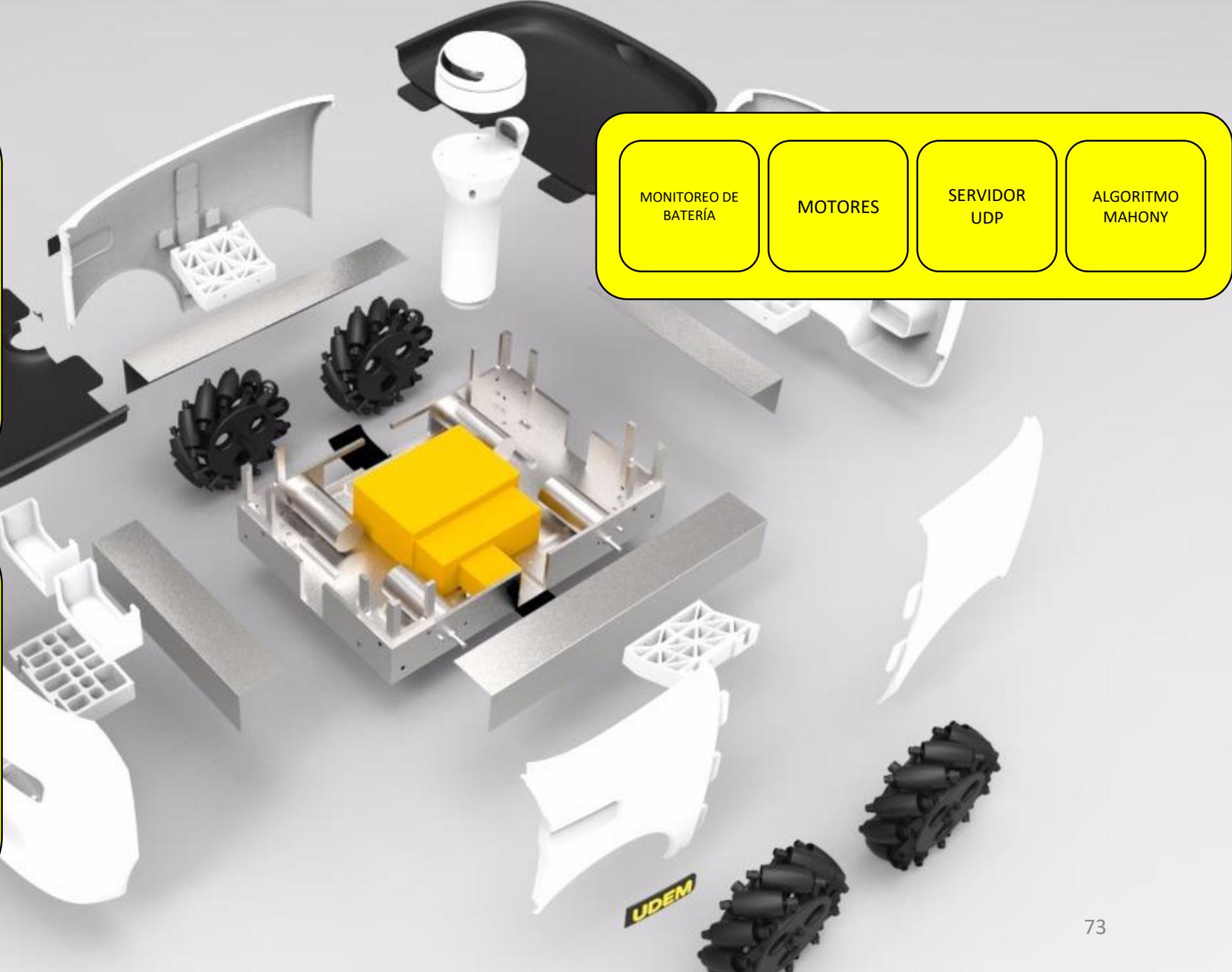
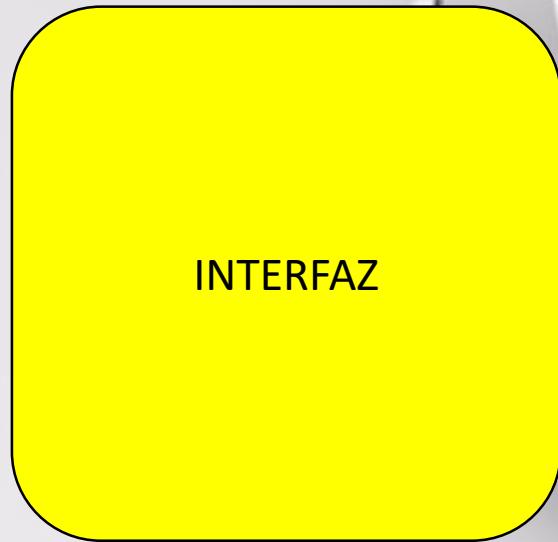
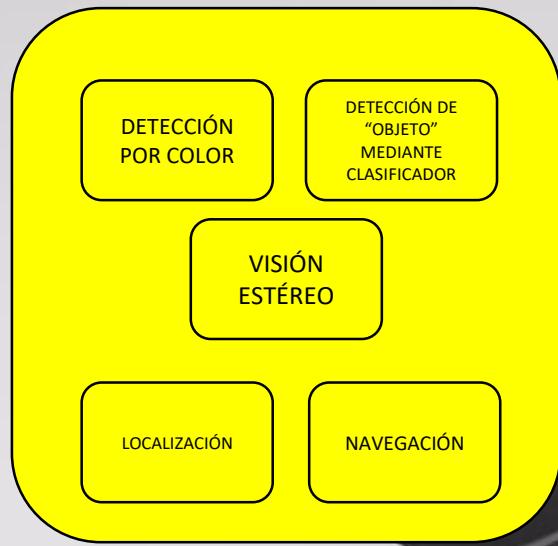
INTERFAZ



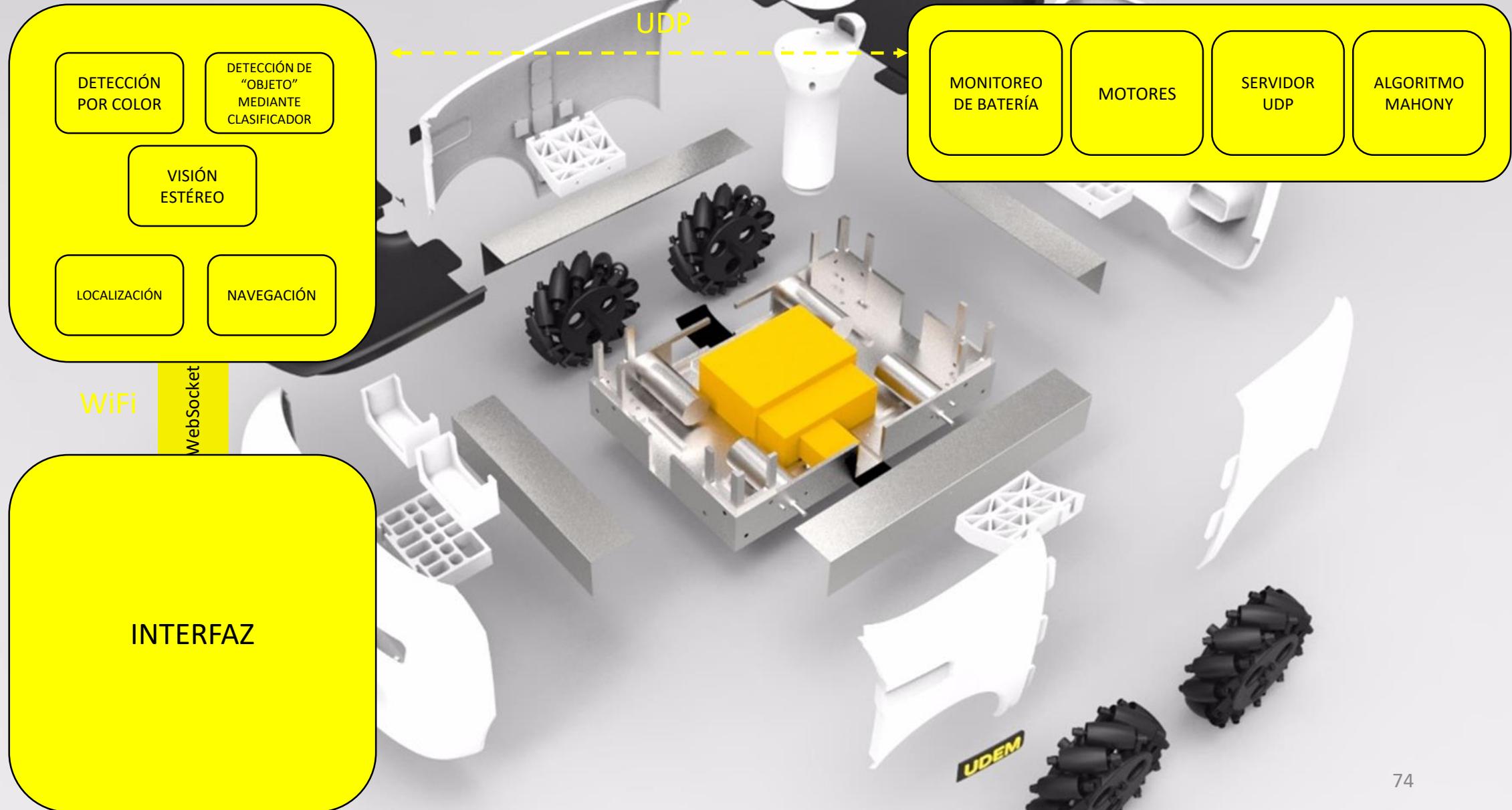
WebSocket

Microsoft Azure



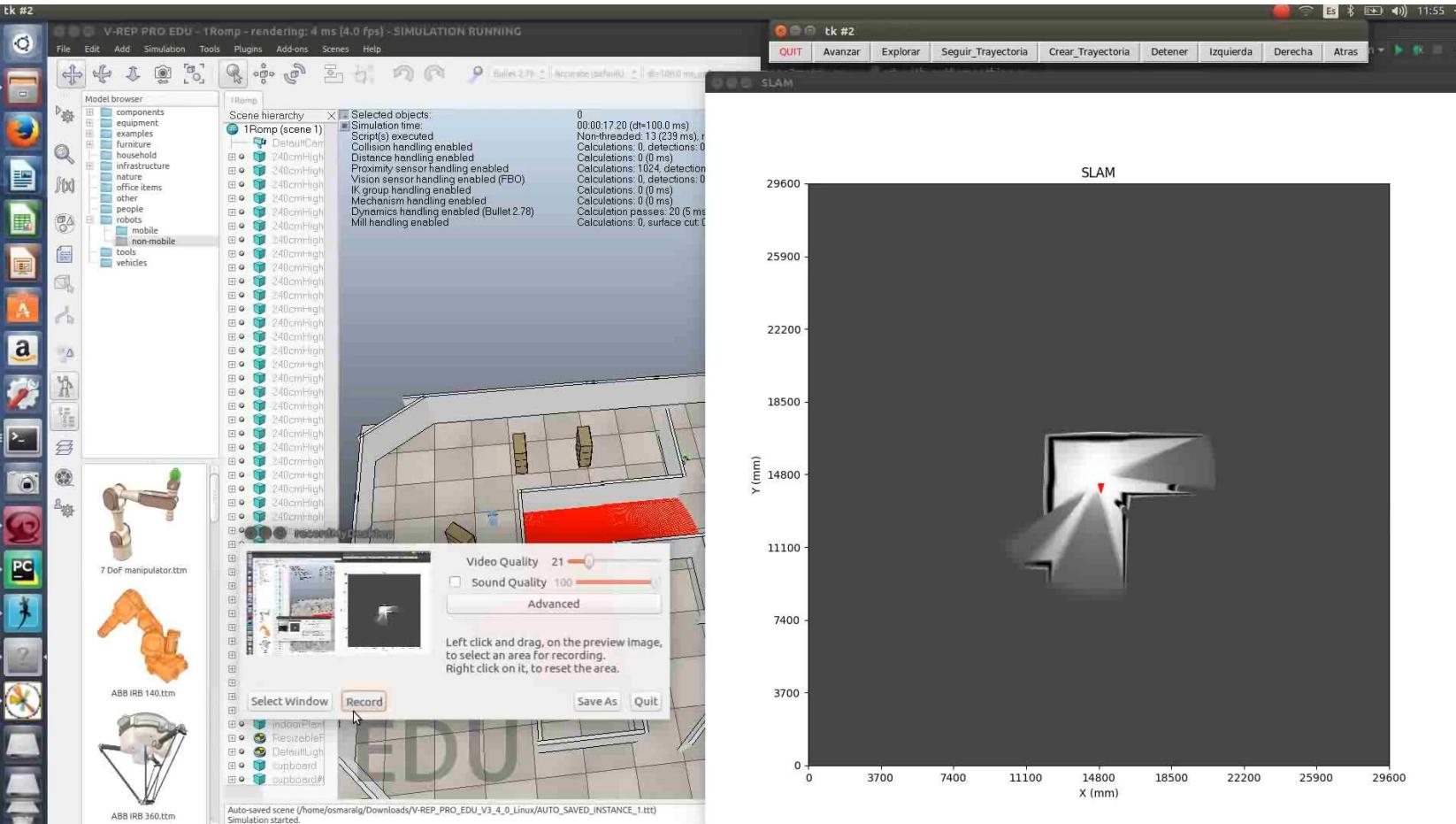


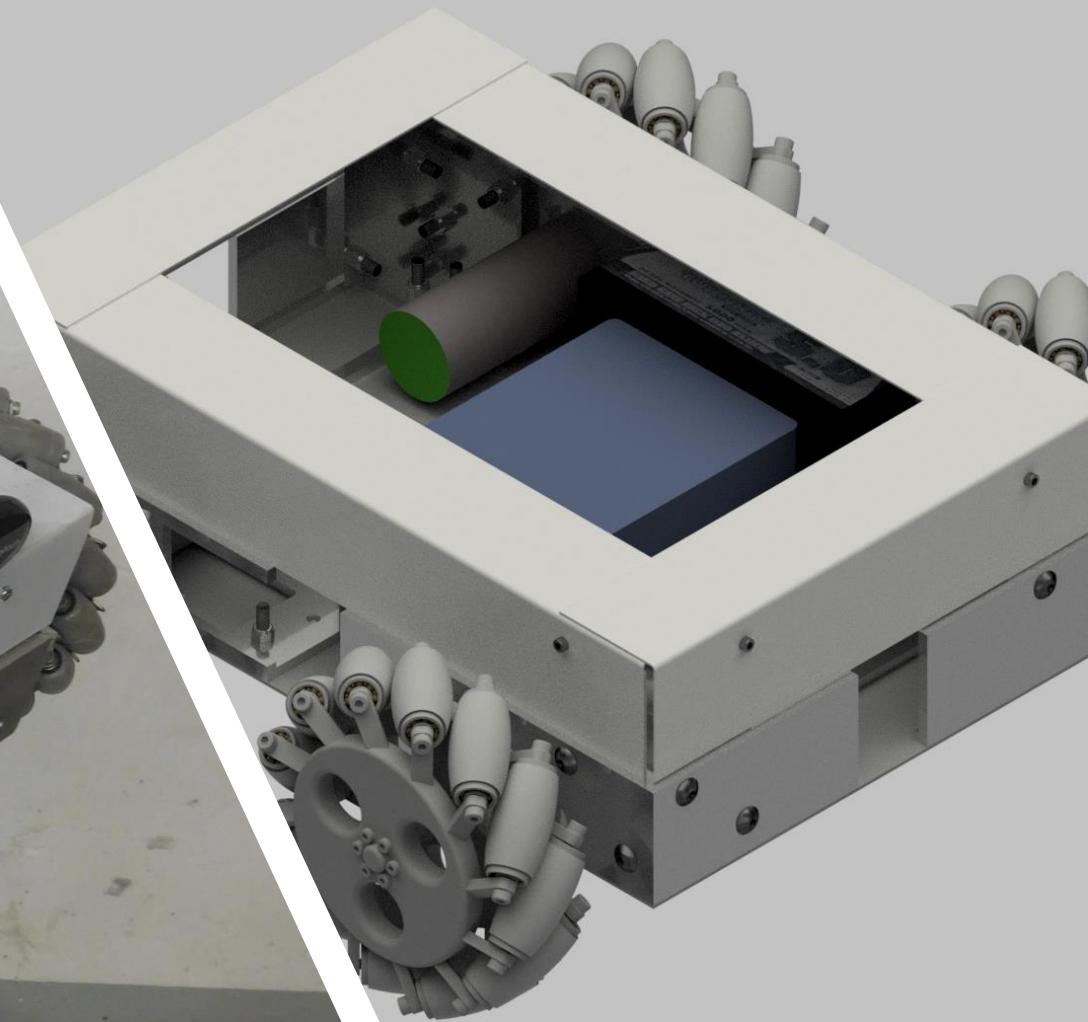
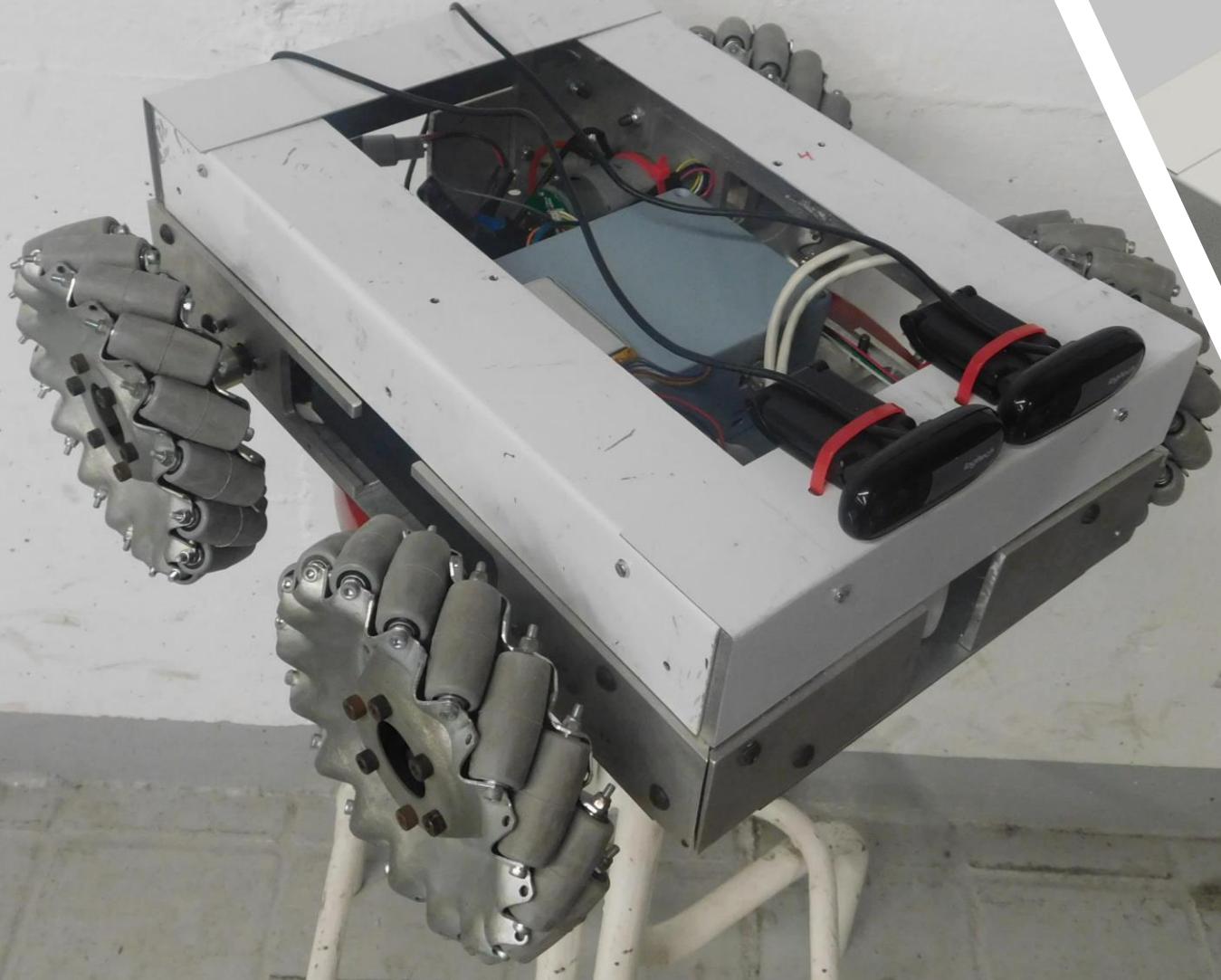
COMUNICACIÓN

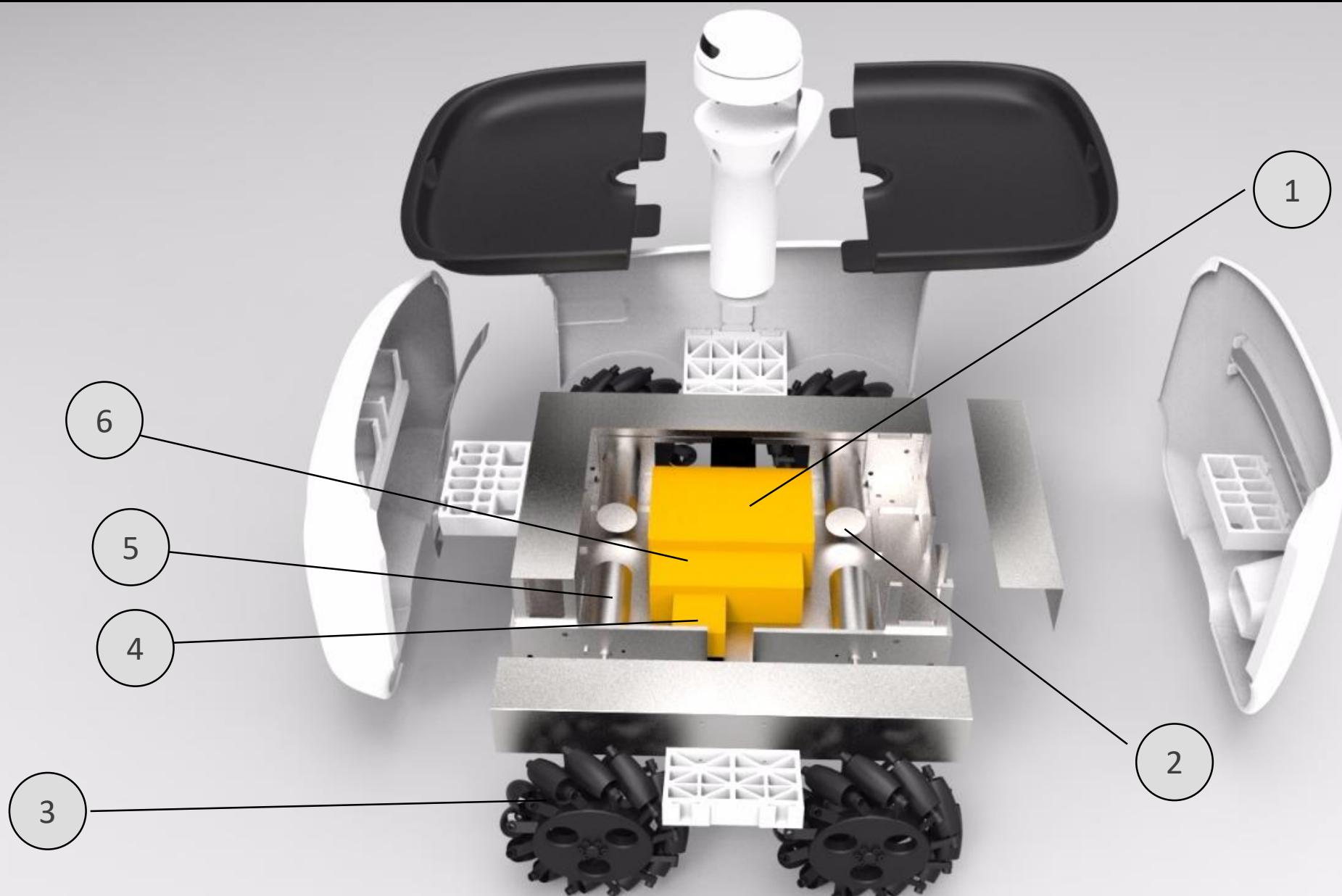


Resultados

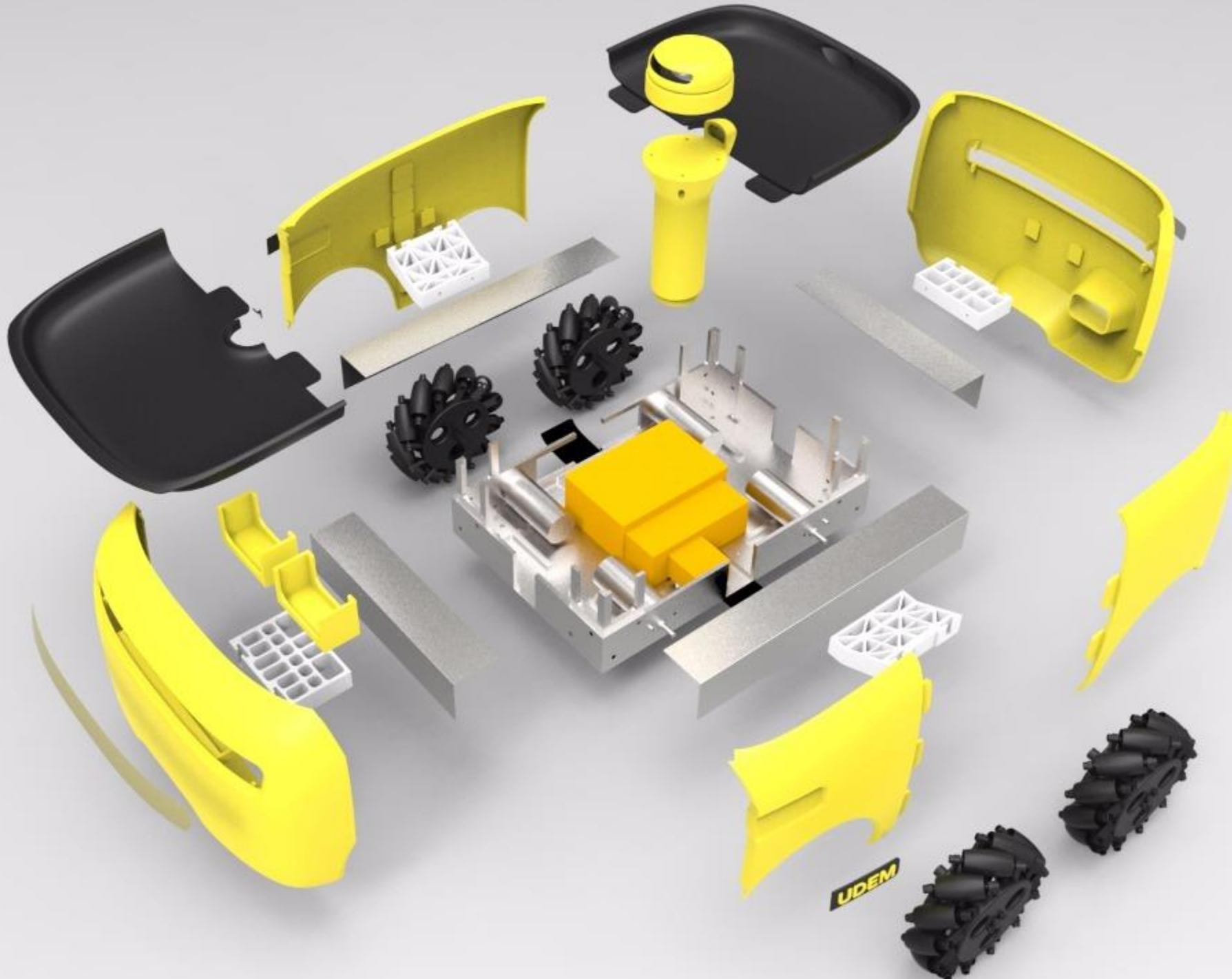
Simulación Final







- 1 MCU
- 2 ENCODER
- 3 OMNI WHEEL
- 4 DRIVER
- 5 MOTOR
- 6 BATERÍA

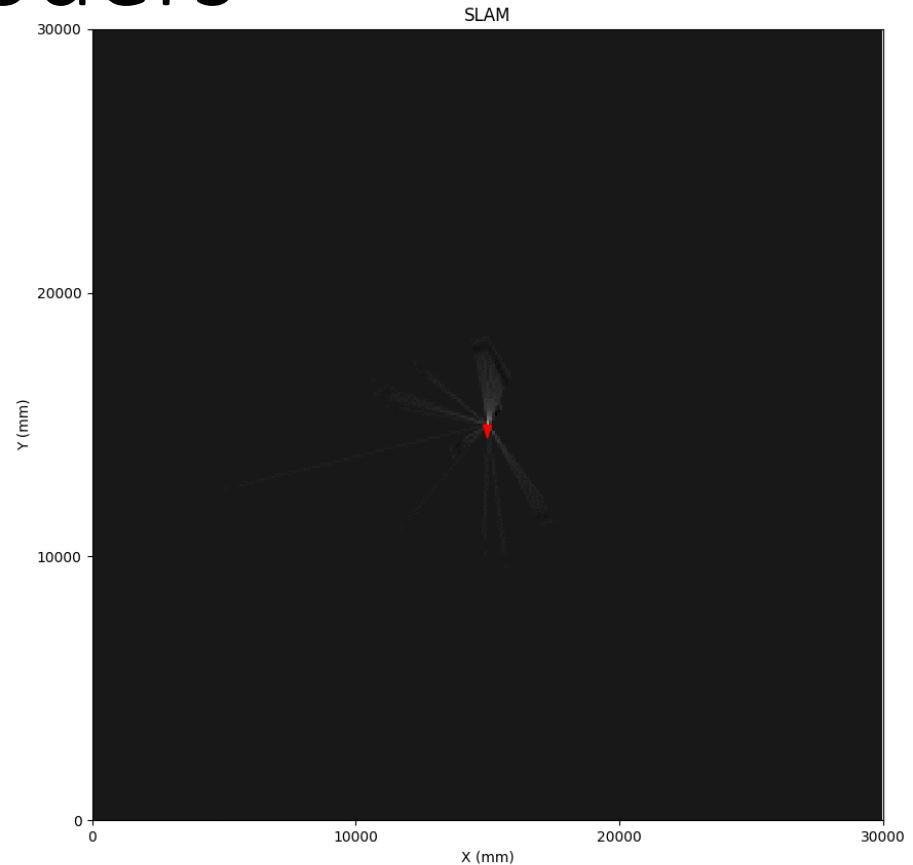






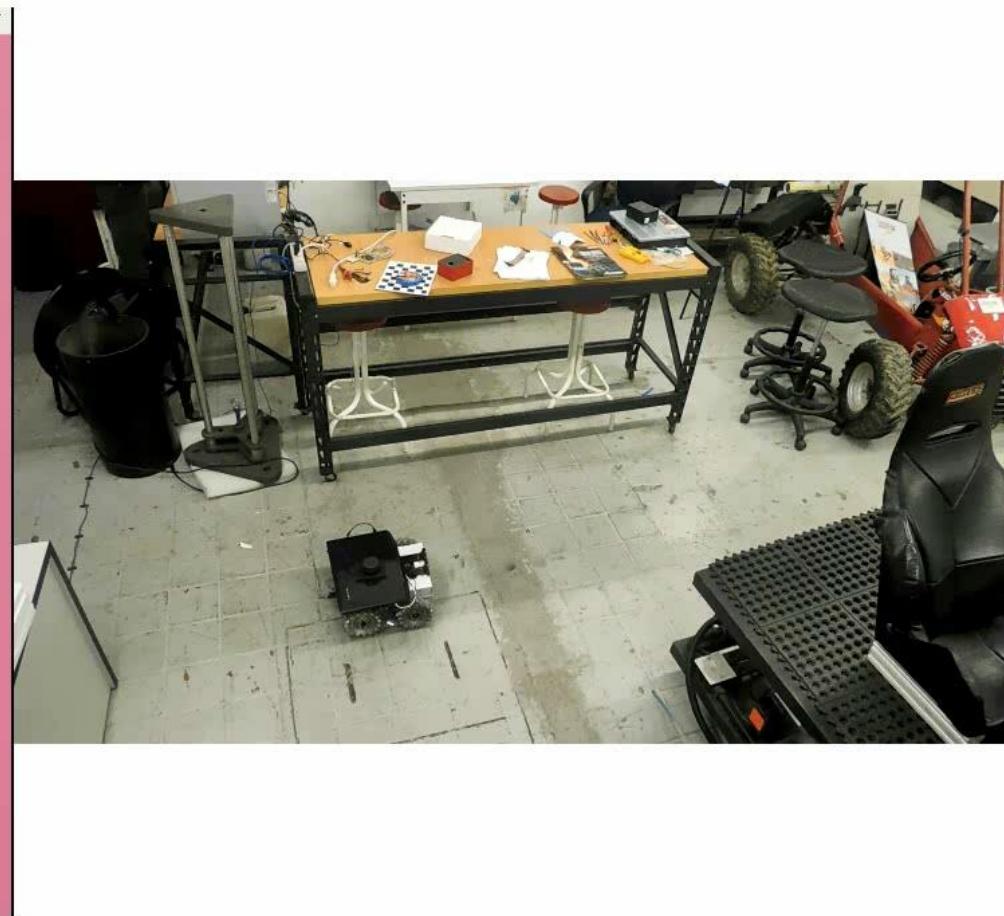
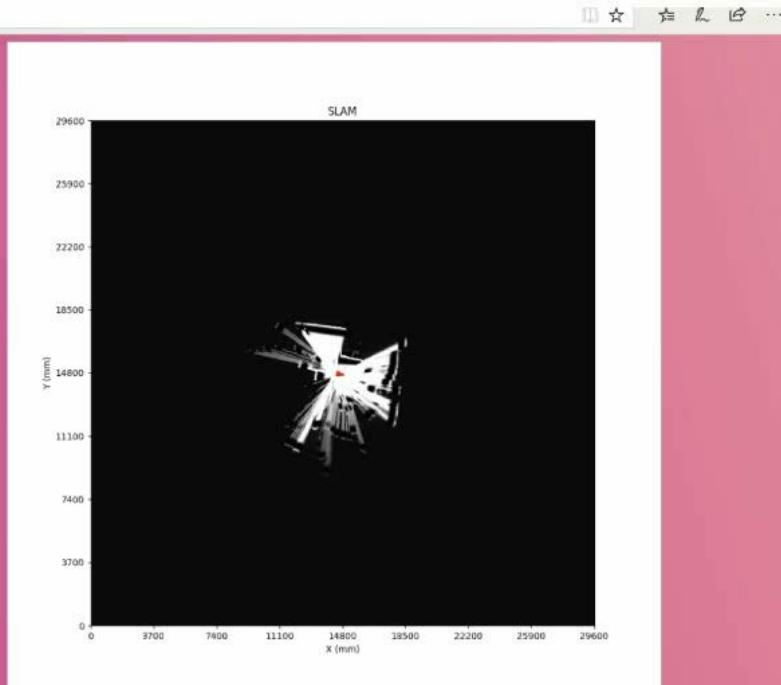
<http://website20171024110201.azurewebsites.net/>

Resultados sistema sin encoders



PEF BOT

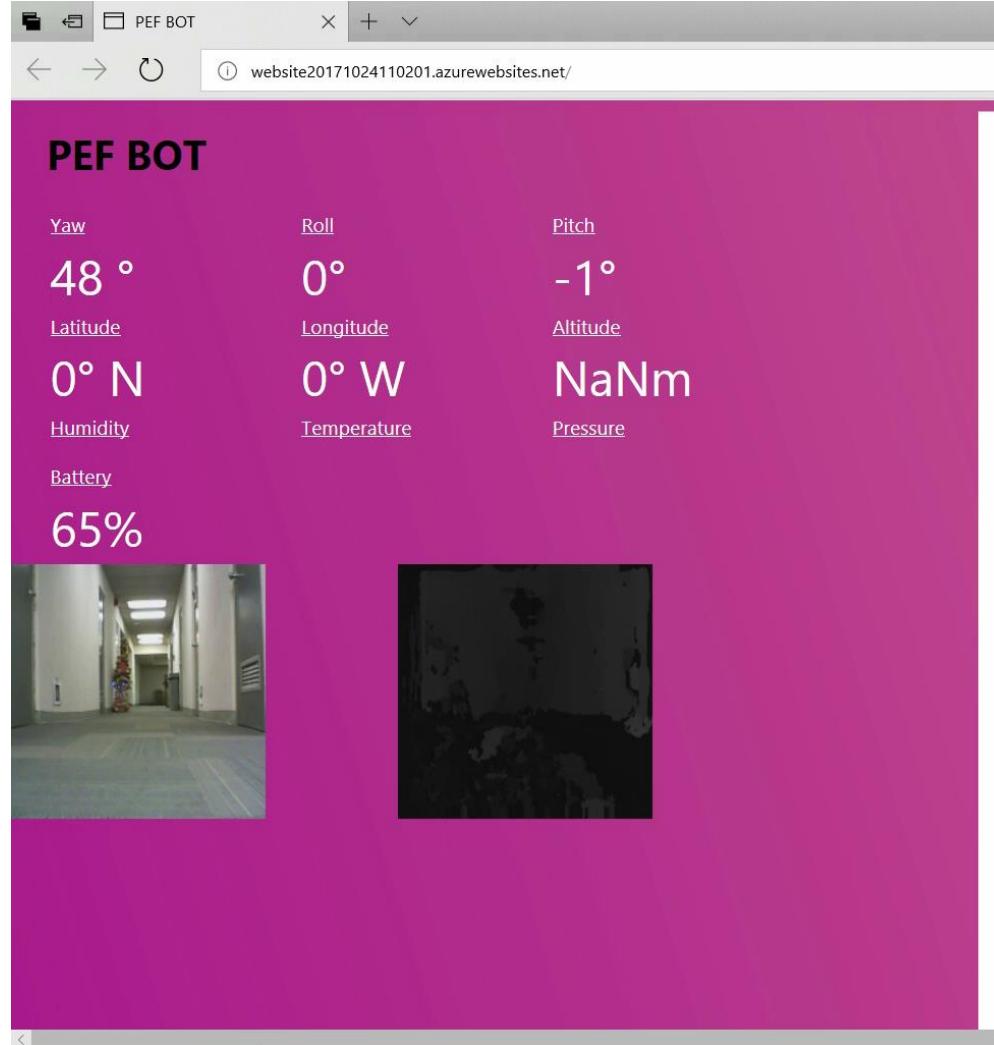
Yaw Roll Pitch
-121 ° 0° 0°
Latitude Longitude Altitude
0° N 0° W NaNm
Humidity Temperature
Battery Pressure
79%
Robot Time
0

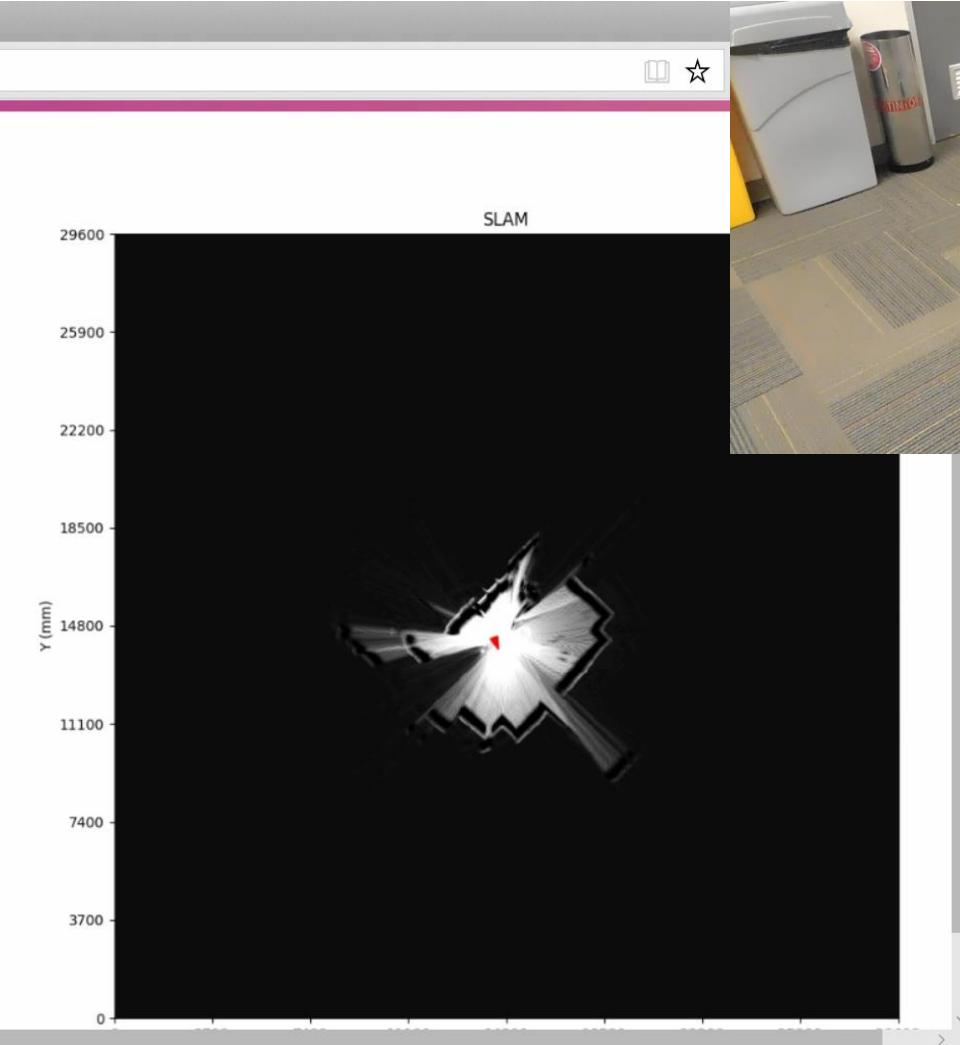


Resultados integración visión

PEF BOT

Yaw	Roll	Pitch
48 °	0°	-1°
Latitude	Longitude	Altitude
0° N	0° W	NaNm
Humidity	Temperature	Pressure
Battery		
65%		







84

.azurewebsites.net/

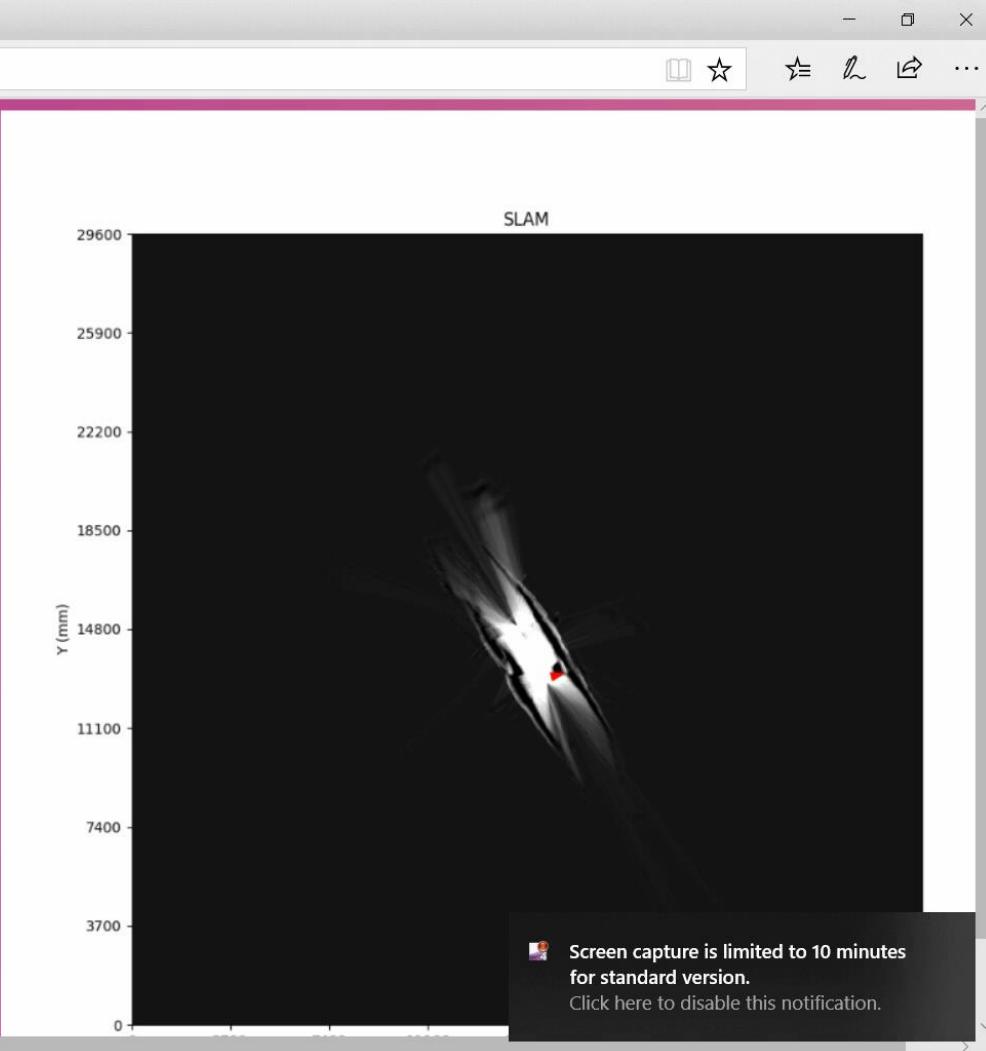
Pitch

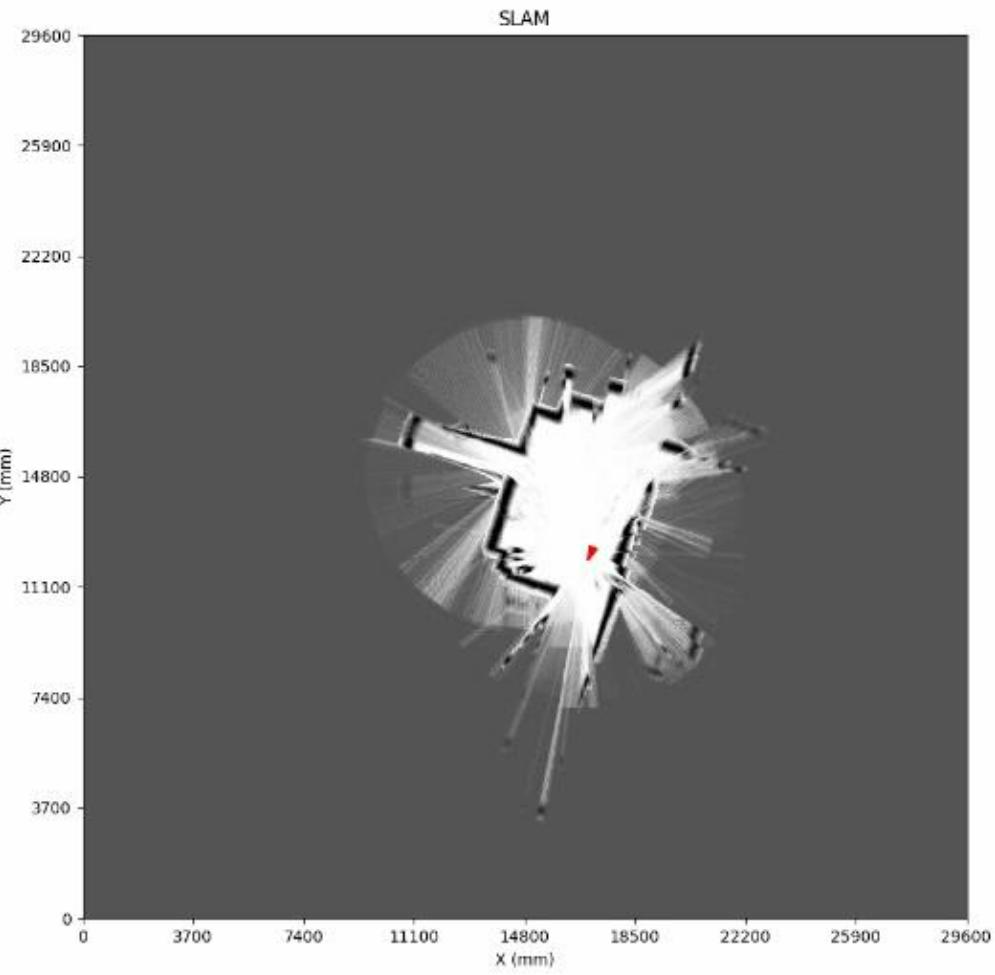
-0°

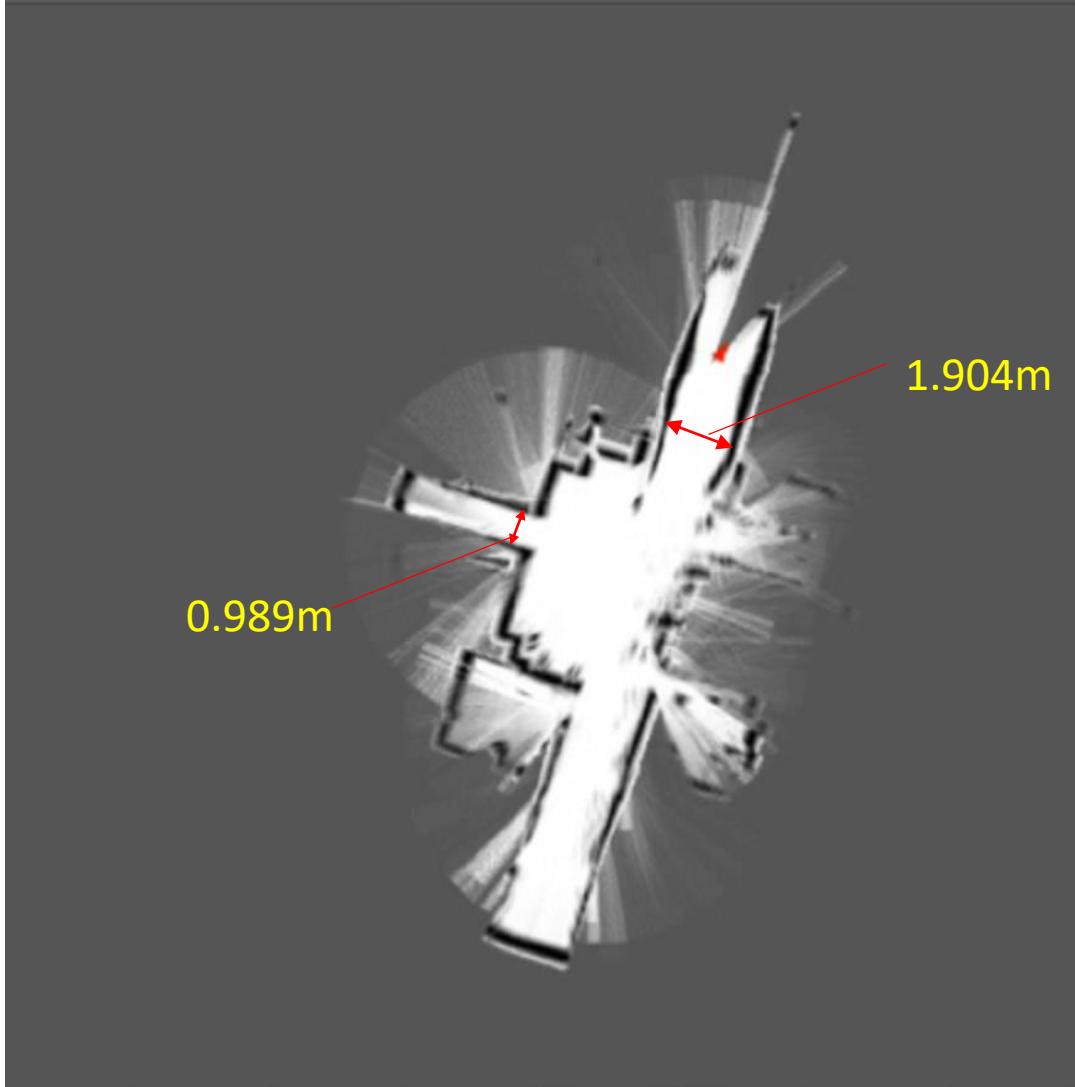
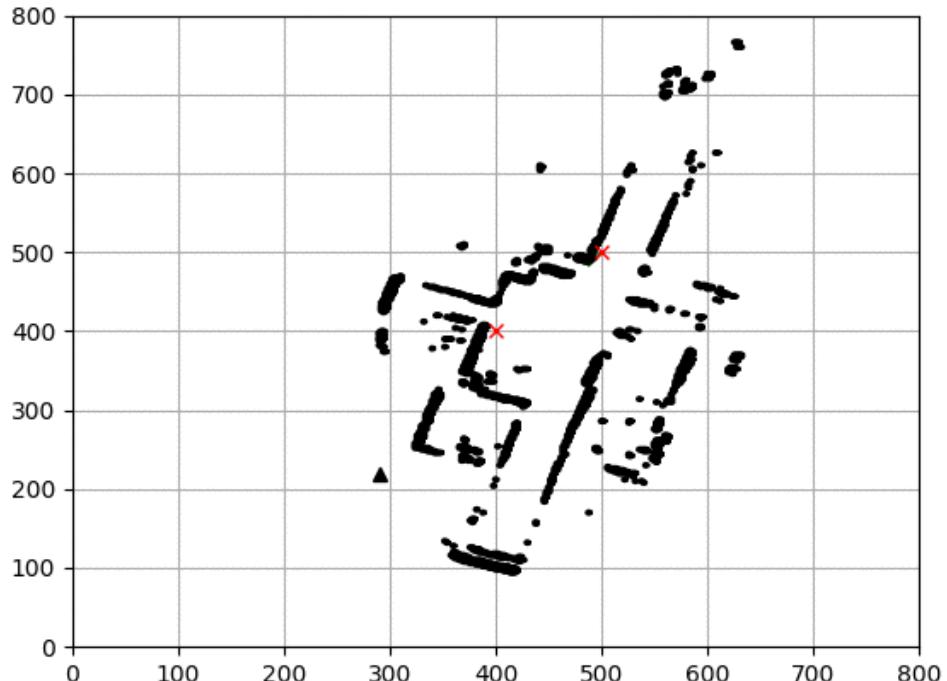
Altitude

NaNm

Pressure







Conclusiones

- Las capacidades de la tarjeta TITech M4 fueron suficientes para el desarrollo del proyecto, cuenta con una gran variedad de periféricos en un espacio de tamaño pequeño y no se descarta su uso en implementaciones futuras. La comunidad de desarrolladores no es comparable con la de Arduino y de otras plataformas abiertas, es desarrollo puramente en lenguaje C habilitando el potencial del microcontrolador.
- La falta de experimentación previa con la unidad de medición inercial (IMU) ocasionó una curva de aprendizaje.
- El filtro complementario tiene un drift aparentemente no lineal. Comprensión del funcionamiento del algoritmo Mahony u otro filtro complementario para una configuración de los sensores y sintonización de variables correcta para una mejor integración en la odometría del vehículo.
- Los encóder aunque ayudan a calcular el desplazamiento, cuando la llanta se patina causan mucho ruido (diferencia entre posición real y estimada).
- La señal del GPS no se puede usar en navegación interior debido a lo débil de la señal.
- Entendimiento de WebSocket para transmisión en tiempo real de video o imágenes e investigación sobre otros protocolos.
- Recolección de datos y análisis de los mismos con un servicio en la nube (Big Data).
- La falta de una computadora dedicada para el procesamiento de imágenes afecta la ejecución de los demás procesos.

Conclusiones

- La simulación de robots (en VREP y MATLAB) nos ayudó a experimentar en el desarrollo temprano del proyecto.
- Un filtro de partículas falla cuando existen dos o más instancias que son buenas estimaciones de las lecturas del Láser.
- Se utilizó un algoritmo de SLAM con filtro de partículas Monte Carlo, sin embargo existe también SLAM con filtro de Kalman y Kalman Extendido que pudiera ayudar a mejorar la estimación de la posición del robot.
- La creación de trayectorias es por medio del algoritmo RRT es rápida sin embargo existen caminos más óptimos pero con mayor demanda computacional.
- El sistema de visión representan los ojos del robot y si hay mucho tiempo de retraso pierde su funcionalidad.
- En la detección de objetos influye mucho la calidad de la cámara y la iluminación del ambiente.
- Considerar la capacidad de cuadros por segundo en la detección para objetos en movimiento.
- Es necesario un procesador dedicado a cada sistema, o mínimo uno al sistema de visión, debido a que afecta la velocidad de transmisión a la interfaz.
- El proceso de compras de la UDEM no fomenta el desarrollo tecnológico.
- Llevar a la práctica nuevos conocimientos es difícil y complicado, sobre todo en la resolución de problemas.

Conclusiones

Se inició con el desarrollo e implementación del SLAM como proyecto en la Universidad de Monterrey. Además se fortaleció el conocimiento de visión por computadora.

Así mismo el proyecto múltiples enfoques como es el de investigación y el de aplicación. El de investigación por creación de investigación nuevo y el de aplicación por la integración de nueva tecnología. Y se crearon pautas para el desarrollo futuro de proyectos similares dentro de la Universidad.