

Robot móvil que monitorea el distanciamiento social obligatorio en aeropuertos

Isidro Rios, Elias
*Escuela profesional de
ingeniería Mecatrónica,
Universidad Ricardo Palma*
Lima, Perú
elias.isidro@urp.edu.pe

Rojas Flores, Alexander
*Escuela profesional de
ingeniería Mecatrónica,
Universidad Ricardo Palma*
Lima, Perú
alexander.rojas@urp.edu.pe

Abstract – This work includes one of the many solutions that exist in the face of this COVID-19 pandemic, which we provide a solution by making a mobile robot for airports to monitor mandatory social distancing, which we use different software for the calculations performed to obtain a kinematic model of a 4-wheeled omnidirectional robot and its programming in Matlab, the mechanical design of the robot and the mecanum-type wheels in the SolidWorks software, and the simulation in CoppeliaSim, as it is a common virtual environment for testing designed robots.

Keywords – Omnidirectional mobile robot, sensors, mandatory social distancing, kinematic model.

Resumen – Este trabajo incluye una de las muchas soluciones que existen ante esta pandemia de COVID-19, la cual brindamos una solución realizando un robot móvil para aeropuertos para monitorear el distanciamiento social obligatorio, el cual utilizamos diferentes software para los cálculos realizados para obtener un modelo cinemático de un robot omnidireccional de 4 ruedas y su programación en Matlab, el diseño mecánico del robot y las ruedas tipo mecanum en el software SolidWorks, y la simulación en CoppeliaSim, ya que es un entorno virtual común para probar robots diseñados.

Palabras claves – Robot móvil omnidireccional, sensores, distanciamiento social obligatorio, modelo cinemático.

I. INTRODUCCIÓN

El 2020 fue el año en que por primera vez en su historia el Aeropuerto Internacional Jorge Chávez cerró sus puertas dada la pandemia causada por la Covid-19, por el tiempo de cuatro meses, el aeropuerto Internacional Jorge Chávez dejó de operar vuelos comerciales llegando a tener consecuencias económicas y sociales.

El Aeropuerto Internacional Jorge Chávez llega a registrar hasta 82 vuelos diarios, de los cuales 15

son para destinos internacionales y los restantes 67 corresponden a destinos nacionales, informó el ministro de Transportes y Comunicaciones, el cual nos da a entender una prospera reactivación de operaciones en este sector y nos enfoca principalmente en la problemática que se desea solucionar en este proyecto.

La solución propuesta para este proyecto es un robot móvil que monitorea el distanciamiento social de las personas en el aeropuerto. La cual el robot se le implementará un sensor de CO2 para poder detectar la aglomeración de personas. Al respetar el distanciamiento social obligatorio existe un menos riesgo de contagios tanto en el lugar donde se encuentra como al lugar donde viajen. El robot también contará con una alerta, comunicando así que las personas que no respeten el distanciamiento social obligatorio puedan cumplirlas.

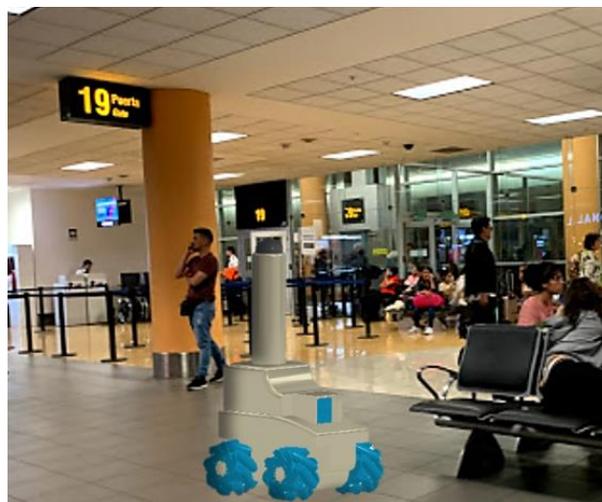


Fig. 1. Robot móvil en el Aeropuerto Jorge Chávez.

II. DISEÑO MECÁNICO

Se realizó el diseño mecánico del robot y las ruedas mecanum, en SolidWorks. El las que se pasó a una distribución larga para el sensor y cámara respectivamente y un chasis simétrico con la finalidad de un acoplamiento ordenado interno de los componentes de control así mismo reduciendo el área de viraje con respecto a sus dimensiones ya que optamos por ruedas mecanum para un omnidireccionamiento cerrado y facilitar la movilidad por su entorno sin mucho espacio.



Fig. 2: Robot móvil diseñado en SolidWorks.

A. Diseño estructural

Chasis diseñado en SolidWorks con material de acero cromado para mejorar su resistencia.

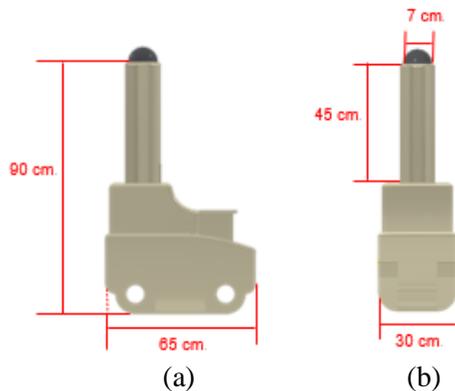


Fig. 3: Medidas de la vista de perfil (a), vista frontal(b) del robot

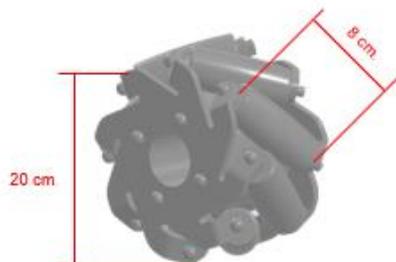


Fig. 4: Medidas de las ruedas Mecanium.

B. Sistema de transmisión mecánica

Definimos la posición del robot considerando el modelo de locomoción diferencial simétrica, en el cual la ubicación de cada rueda mecanium fue posicionada con respecto al centro de gravedad de nuestro robot, para repartir el peso total y tener una mejor distribución de reacciones con respecto a la carga total de todo el chasis del robot.

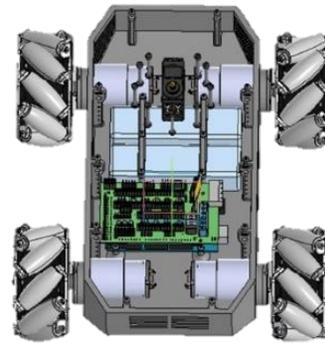


Fig. 5: Soporte mecánico del chasis.

III. DISEÑO ELÉCTRICO Y ELECTRÓNICO

Los materiales a usar son, 2 motorreductores para el movimiento del robot, un sensor ultrasónico para que el robot no choque con cualquier obstáculo que se le presente, una cámara VC0706 que servirá el monitoreo y desplazamiento dentro del aeropuerto, un sensor de CO2, un protoboard, resistencias de 1k y 330 ohmios, un circuito integrado L293D para controlar los motores, un servo motor para mover el sensor ultrasónico y una pantalla LCD de 16x2 para la visualización del censado.

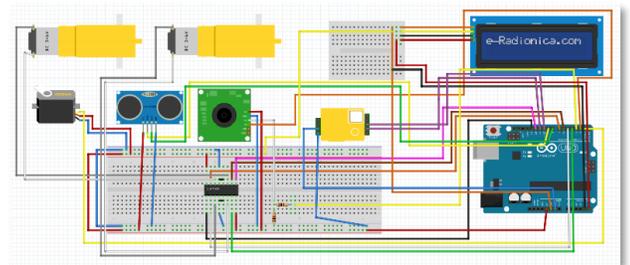


Fig. 6: Circuito para el Robot móvil diseñado en el software Fritzing.

IV. ANÁLISIS Y SIMULACIÓN CINEMÁTICA

Existen varios tipos de robots móviles terrestres, se suelen clasificar según su tipo de movimiento. Entre los que usan ruedas los 2 más comunes son el uni-ciclo y el omnidireccional, además de ello otra característica es el tipo de rueda escogido para el proyecto, el cual será "mecanum".

A. Cálculos

Por su capacidad de desplazarse en cualquier dirección sin la necesidad de alcanzar previamente una orientación específica en toda clase de entornos superficiales, se realizaron los cálculos del modelo de cinemática del robot omnidireccional de 4 ruedas. La característica de este tipo de robot, es que a diferencia de la mayoría de móviles, cuenta con 2 velocidades, comúnmente llamadas frontal y lateral, pues su tipo de llanta peculiar le permite hacer giros sin necesidad de orientarse previamente.

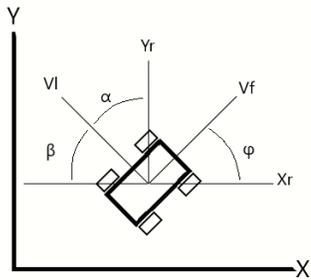


Fig. 7: coordenadas del robot en el plano XY

$$Xr = Xc$$

$$Yr = Yc$$

$$Xr' = Xc'$$

$$Yr' = Yc'$$

$$Xc' = Vf \cdot \cos(\varphi) - Vl \cdot \sin(\alpha)$$

$$Yc' = Vf \cdot \sin(\varphi) + Vl \cdot \cos(\alpha)$$

$$\beta + \alpha = 90^\circ$$

$$\varphi + 90 + \beta = 180^\circ$$

$$\beta = 90^\circ - \varphi$$

$$\alpha = \varphi$$

Modelo cinemático

$$Xr' = Vf \cdot \cos(\varphi) - Vl \cdot \sin(\varphi)$$

$$Yr' = Vf \cdot \sin(\varphi) + Vl \cdot \cos(\varphi)$$

$$\varphi' = \omega$$

Modelo matricial

$$\begin{bmatrix} Xr' \\ Yr' \\ \varphi' \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Vf \\ Vl \\ \omega \end{bmatrix}$$

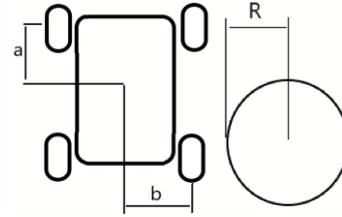


Fig. 8: Dimensiones de las ruedas del robot.

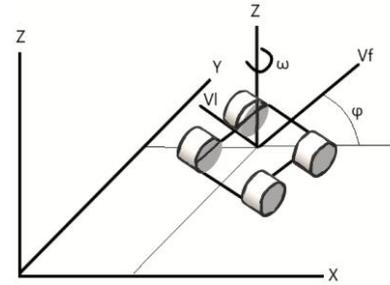


Fig. 9: Relación de ruedas y velocidades.

$$\begin{bmatrix} Wd1 \\ Wd2 \\ Wd3 \\ Wd4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} Vf \\ Vl \\ \omega \end{bmatrix}$$

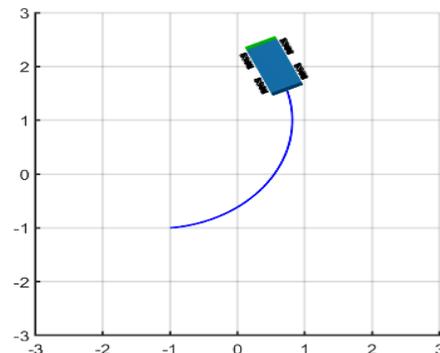


Fig. 10: Trayectoria realizada por el Robot.

V. DISEÑO DE SOFTWARE Y SIMULACIÓN

Los cálculos obtenidos a través del modelo cinemático son necesarios para poder controlar el robot. Según lo visto anteriormente se puede mover un robot, dando velocidades lineales y velocidad angular. Para probar el desenvolvimiento del modelo, se utilizó el software Matlab, por su capacidad de procesar cálculos matriciales.

```

% Cinemática de un robot móvil omnidireccional
clc
clear all
close all

% Tiempo de simulación
ts=0.1;
t=0:ts:20;

%Velocidades
vf=0.0*ones(1,length(t));
vl=0.0*ones(1,length(t));
w=-0.0*ones(1,length(t));

% Posición de llantas respecto al centro
a=0.15;
b=0.10;

% Radio de la llanta
R=0.02;

T=(1/R)*[1 -1 -(a+b);...
          1 1 (a+b);...
          1 1 -(a+b);...
          1 -1 (a+b)];

% Posición inicial
xr(1)=0;
yr(1)=0;
phi(1)=0;

for k=1:length(t)

    % Movimiento del robot
    xrp(k)=vf(k)*cos(phi(k))-vl(k)*sin(phi(k));
    yrp(k)=vf(k)*sin(phi(k))+vl(k)*cos(phi(k));

    Wd=T*([vf(k);vl(k);w(k)]);

    Wd1(k)=Wd(1);
    Wd2(k)=Wd(2);
    Wd3(k)=Wd(3);
    Wd4(k)=Wd(4);

    % Integración numérica
    xx(k+1)=xr(k)+ts*xrp(k);
    yy(k+1)=yr(k)+ts*yrp(k);
    phi(k+1)=phi(k)+ts*w(k);

end

pasos=10; fig=figure;
fig.Color = [1 1 1];
set(gca,'FontSize',16)
set(gca,'linewidth',2)
set(fig,'position',[10 60 980 600]);
axis square; cameratoolbar
grid on;
axis([-3 3 -3 3 -0.2 1]); grid on
camlight right % scene light

Dimension_Omni;

M1=Plot_Omni(xr(1),yr(1),phi(1));hold on
M2 = plot(xr(1),yr(1),'b','linewidth',2);

for i=1:pasos:length(t)

    delete (M1)
    delete (M2)
    M1=Plot_Omni(xr(i),yr(i),phi(i)); hold on
    M2=plot(xr(1:i),yr(1:i),'b','LineWidth',2);

    pause(ts)
end

```

Fig. 11: Código en Matlab

```

#include <Wire.h>
#include <Arduino.h>
#include <MHZ19.h>
#include <Adafruit_VCO706.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

Servo servo;

#define RX_PIN 10
#define TX_PIN 11
#define BAUDRATE 9600

MHZ19 myMHZ19;
SoftwareSerial mySerial(RX_PIN, TX_PIN);
LiquidCrystal_I2C lcd(0x3F,16,2);

int pos;
int puenteR = 13;
int motorRF = 6;
int motorRB = 7;
int puenteL = 12;
int motorLF = 4;
int motorLB = 5;
unsigned long pulso;

float distancia;
float derecha;
float izquierda;

int trig = 9;
int eco = 8;
int duracion = 0;

duracion = pulseIn(eco, HIGH);
distancia = (duracion/2)/28.5;
Serial.println(distancia);
delay(100);

if(distancia < 35)
{
    digitalWrite(puenteR, LOW);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorRB, LOW);
    digitalWrite(puenteL, LOW);
    digitalWrite(motorLF, LOW);
    digitalWrite(motorLB, LOW);

    servo.write(pos = 0);
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);
    duracion = pulseIn(eco, HIGH);
    derecha = (duracion/2)/28.5;
    delay(1000 + izquierda);
    Serial.print("derecha =");
    Serial.print(derecha);

    servo.write(pos = 180);
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);
    duracion = pulseIn(eco, HIGH);
    izquierda = (duracion/2)/28.5;
    delay(1000 + izquierda);
    Serial.print("izquierda =");
    Serial.println(izquierda);

    delay(400);
}
else
{
    servo.write(pos = 90);
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);
    duracion = pulseIn(eco, HIGH);
    distancia = (duracion/2)/28.5;
    digitalWrite(puenteR, HIGH);
    digitalWrite(motorRF, LOW);
    digitalWrite(motorRB, HIGH);
    digitalWrite(puenteL, HIGH);
    digitalWrite(motorLF, HIGH);
    digitalWrite(motorLB, LOW);

    delay(400);

    if(derecha > izquierda)
    {
        digitalWrite(puenteR, HIGH);
        digitalWrite(motorRF, HIGH);
        digitalWrite(motorRB, LOW);
        digitalWrite(puenteL, HIGH);
        digitalWrite(motorLF, HIGH);
        digitalWrite(motorLB, HIGH);
        delay(400);
    }
    else
    {
        digitalWrite(puenteR, HIGH);
        digitalWrite(motorRF, LOW);
        digitalWrite(motorRB, HIGH);
        digitalWrite(puenteL, HIGH);
        digitalWrite(motorLF, HIGH);
        digitalWrite(motorLB, LOW);

        digitalWrite(puenteL, HIGH);
        digitalWrite(motorLF, LOW);
        digitalWrite(motorLB, HIGH);
        delay(1000);
    }
}

static int CO2=0;
CO2 = myMHZ19.getCO2();
Serial.print("CO2 (ppm): ");
lcd.print("CO2 (ppm):");
Serial.println(CO2);
lcd.print(CO2);
delay(1000);
}

void setup() {
    pinMode ( trig, OUTPUT);
    pinMode ( eco, INPUT);
    pinMode ( puenteR, OUTPUT);
    pinMode ( motorRF, OUTPUT);
    pinMode ( motorRB, OUTPUT);
    pinMode ( puenteL, OUTPUT);
    pinMode ( motorLF, OUTPUT);
    pinMode ( motorLB, OUTPUT);

    digitalWrite(puenteR, LOW);
    digitalWrite(puenteL, LOW);

    servo.attach(1);

    lcd.init();
    lcd.backlight();

    Serial.begin(9600);
    mySerial.begin(BAUDRATE);
    myMHZ19.begin(mySerial);

    myMHZ19.autoCalibration();
}

void loop() {
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);
    servo.write(pos = 90);
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);
    duracion = pulseIn(eco, HIGH);
    distancia = (duracion/2)/28.5;
    delay(1000);
}

```

Fig. 12: Código en Arduino para el Robot móvil.

VI. PRUEBAS Y RESULTADOS

Las pruebas de nuestro diseño de hardware y programación fueron realizadas en el programa CoppeliaSim. Así mismo se pasó a observar el cumplimiento de nuestra cinemática tanto teórica como practica en un ambiente virtual que sea asemejaría a un ambiente real en el aeropuerto de lima.

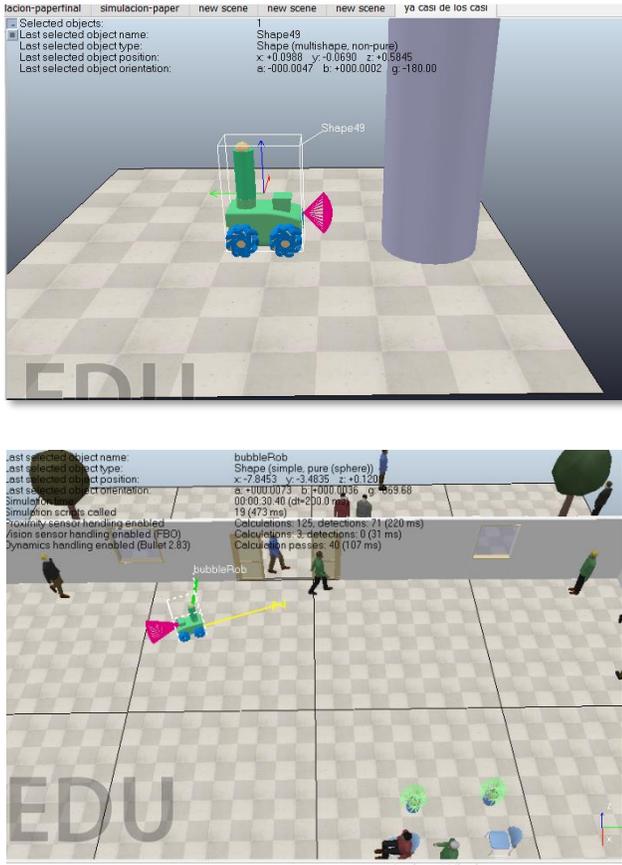


Fig. 13: Pruebas de movimiento del Robot móvil en CoppeliaSim.

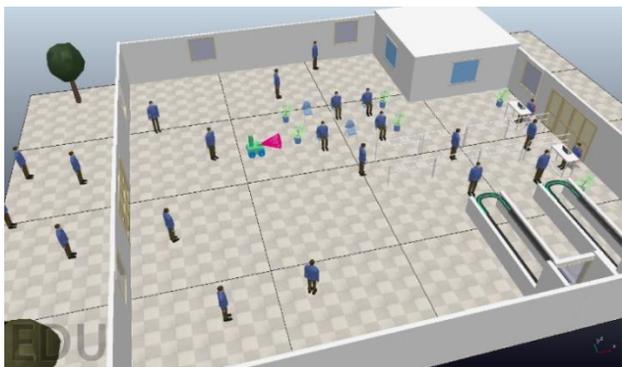


Fig. 14: Robot móvil en CoppeliaSim en el aeropuerto virtual

VII. PRUEBAS Y RESULTADOS

El presente trabajo muestra el desarrollo y simulación de un robot orientado a la seguridad de las personas por medio de un diseño móvil y unos sensores periféricos así mismo con una cámara, para alertar las medidas de bioseguridad en el distanciamiento de los aeropuertos.

Los diseños fueron realizados tanto en inventor como el SolidWorks, para tener un mejor alcance tanto en simulación que sea compatible con el diseño realizado y así mismo en la programación para recolección de datos cinemáticos, ya que estos dos programas son universales en compatibilidad de simulación e implementación.

Se paso a desarrollar lo aprendido en el curso de acuerdo a lo establecido con las competencias desarrolladas en cinemática de un robot así mismo su dinámica y movilidad con el objetivo de demostrar los laboratorios y las teorías recibidas.

AGRADECIMIENTOS

Gracias a la escuela Profesional de Ingeniería Mecatrónica de la Universidad Ricardo Palma, Lima, Perú. Además, gracias por la supervisión de nuestro profesor Palomares Orihuela, Ricardo John en el curso de Modelamiento de Robots.

REFERENCIAS

- [1] A. Saénz, V. Santibáñez y E. Bugarin, *Control de velocidad de un robot omnidireccional con dinámica de actuadores*, *AMRob Journal, Robotics: Theory and Applications*, 2016.
- [2] R. S. Santiago, *Control y Comportamiento de Robots Omnidireccionales*, Universidad de la República Montevideo - Uruguay, 2009.
- [3] M. I. Aguilera, M. A. bautista y J. Iruegas, *Diseño y Control de Robots Móviles*, Instituto Tecnológico de Nuevo Laredo Reforma Sur 2007.
- [4] Silva Ortigoza, R.; García Sánchez, J. R.; Barrientos Sotelo, V. R.; Molina Vilchis, M. A.; Hernández Guzmán, V. M.; Silva Ortigoza, G. *Una panorámica de los robots móviles Télématique*, vol. 6, núm. 3, 2007, pp. 1-14 Universidad Privada Dr. Rafael Belloso Chacín Zulia, Venezuela.
- [5] Barrientos Sotelo, Víctor Ricardo; García Sánchez, José Rafael; Silva Ortigoza, Ramón *Robots Móviles: Evolución y Estado del Arte Polibits*, núm. 35, 2007, pp. 12-17 Instituto Politécnico Nacional Distrito Federal, México.