

# SISTEMA SEMI-AUTOMATICO DE RECOJO DE RESIDUOS PARA HOGARES EN DISTRITOS CON ZONAS PLANAS

Ralf G. Vivanco Monteza  
 Universidad Ricardo Palma, Lima,  
 Perú  
 Email: ralf\_5395@hotmail.com

Abel A. H. W. L. Arbildo Jurupe  
 Universidad Ricardo Palma, Lima,  
 Perú  
 Email: abel\_john\_14@hotmail.com

Gerson M. Collantes Pizarro  
 Universidad Ricardo Palma, Lima,  
 Perú  
 Email:gersonmanuelcp@gmail.com

*Abstract—Lima is flooded with garbage, this is something worrisome because citizens grow up with the wrong idea of having garbage in the streets is normal, this project tries to generate awareness among citizens if it is true not to revolutionize the idea of a neat file but if you push for change, the purpose of the project is to apply the knowledge acquired in the mechatronic engineering career to develop the project "automated garbage truck" so in a novel way to begin the change of taking the garbage out of our streets.*

*The levels of garbage in cities are increasing. Generally, the greater the economic development and the degree of urbanization has an urban nucleus, the more waste it produces.*

**Palabras Claves—Automatización, Residuos, Camión, Diseño, Control.**

## I. INTRODUCCION

Lima está inundada de basura, esto es algo preocupante por el hecho que los ciudadanos creemos con la idea equivocada de tener basura en las calles es normal, este proyecto trata de generar conciencia entre los ciudadanos si es verdad no revolucionar la idea de una lima pulcra pero si impulsar al cambio, el fin del proyecto es aplicar los conocimientos adquiridos en la carrera ingeniería mecatrónica para desarrollar el proyecto “camión de basura automatizado” para así de una manera novedosa comenzar el cambio de sacar la basura de nuestras calles.

Los niveles de basura en las ciudades van en aumento. Generalmente, cuanto mayor desarrollo económico y grado de urbanización tiene un núcleo urbano, más cantidad de desechos produce. Actualmente se generan 1,3 miles de millones de residuos sólidos urbanos al año y se espera que esta cifra se duplique en 2015, según informa el Banco Mundial [1].



Figura1: Residuos en la ciudad limeña

## SISTEMAS DE RECOGIDA:

Dependiendo del sistema utilizado la la realización de la pre-recogida, así será la recogida de residuos en cuanto a las

necesidades de vehículos y sistemas de trabajo. La Recogida de residuos puede clasificarse de la siguiente manera [2]:

RECOGIDA	Con vehículo	Sin compactación	Recogida manual	Manual
		Con compactación	Recogida trasera	Manual
			Recogida lateral	Mecanizada
	Sin vehículo	Via seca	Recogida neumática	Automatizada
		Via húmeda	Triturador	Otros

Figura2: Cuadro de clasificación sobre el sistema de recogida de residuos

El sistema que se usará en el proyecto será el de recojo lateral, este sistema consiste en que los contenedores son recogidos lateralmente por un elevador que acciona el conductor con la ayuda de 4 o 5 cámaras y un televisor de una o dos pantallas[3].



Figura3: Contenedor de carga lateral (3200 litros), y de servicio con camión recolector-compactador de carga lateral / Diseño en SolidWorks del contenedor usado en el proyecto

Por otro lado, también se presentan algunas herramientas que son útiles para dar forma a la solución, esto es, generar un modelo funcional que sea capaz de clasificar diferentes residuos sólidos con la ayuda de un Controlador de Lógica Programable (Programmable Logic Controller, PLC). Se introduce la noción de lo que es un PLC, cómo trabaja y cuál es su estructura de programación. Se aplica una metodología de diseño que permite generar el concepto del Clasificador Automático de Residuos Sólidos (CARS) [4].

Esta es una de las posibles soluciones que a grande escala son aplicables, para nuestro proyecto para manera de práctica se utilizara la plataforma arduino.

Para sustituir el sistema tradicional de recojo de basura en las ciudades especialmente en las zonas planas de nuestra ciudad se pensó en el diseño de este proyecto como ya antes mencionado incentivar a población a tener una cultura de limpieza en cuanto a nuestras calles, puesto que es muy importante porque esto generan problemas graves como es la contaminación en el medio ambiente, contaminación que

cada vez es más crítica a nivel mundial. Y este proyecto es como la aportación en una pequeña pero no despreciable interesante ante el caos de la eliminación de los residuos en las grandes ciudades.

## II. SISTEMA MECATRÓNICO

Con la finalidad de elaborar un prototipo del camión automatizado para la recolección de basura en las ciudades que tengan como característica sus zonas planas que funcione eficientemente y soluciones los inconvenientes que genera el desorden de la eliminación de los desechos en las ciudades ya mencionadas, no enfocaremos en cubrir las siguientes necesidades:

- Reducir los accidentes en el proceso.
- Obtener una regularización parcial de los residuos.
- Bajo costo de maquinaria.
- Fácil instalación.

### A. Materiales

- Motores DC
- Arduino
- MDF de 0.3mm (material base para parte mecánica)
- Sensor infrarrojo
- Sensor ultrasonido
- Resistencias y capacitores
- Modulo Puente H

Con los materiales propuestos y las recomendaciones, se implementó el circuito para el control de los motores como se muestra en la figura:

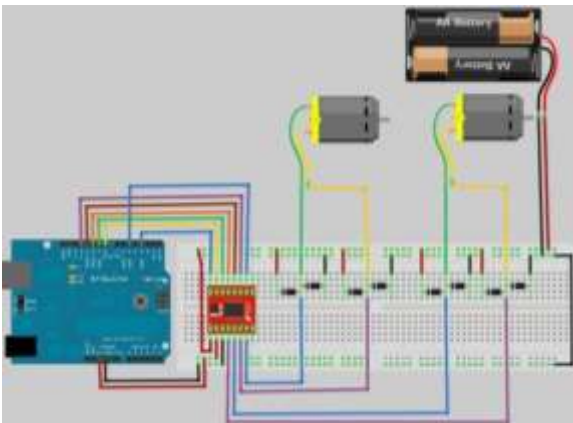


Figura 4: Esquema de conexión del puente h y los motores con arduino

El TB6612FNG es un controlador de puente H - no entraré en los detalles técnicos de *Cómo* se trabaja, pero el resultado final es que se puede convertir un motor conectado en cualquier dirección y a una velocidad variable. Esto le permite construir un robot muy ágil impulsado por 2 motores que pueden:

- seguir adelante
- moverse hacia atrás
- girar en un círculo muy apretado por conducir un motor adelante y uno hacia atrás
- una variedad de otros movimientos con velocidades motor diferencial y direcciones

Para mantener este instructable centrado en el conductor, ha construido el bot en un simple cuerpo de Mecano y no han

extendido para incluir cualquier detección de obstáculos - el bot sólo conduce a un patrón determinado.

Para satisfacer estas necesidades, el sistema Mecatrónica debe cumplir con los siguientes requerimientos mecánicos, eléctricos y de control:

### Requerimientos mecánicos:

- El material para la estructura debe de ser de bajo costo y con poco peso.
- Debe ser construido con materiales no tóxicos.
- Diseño realizado en software SOLIDWORK.



Figura 5: Diseño realizado en SolidWork

### Requerimientos electrónicos:

- El suministro de energía eléctrica del sistema será de una toma 6V.
- Debe tener un consumo bajo energético.
- Tener un sistema de protección y aislamiento eléctrico para proteger el sistema de control.

### Requerimientos de control:

- Uso de un microcontrolador capaz de controlar sin problemas todos los sensores y actuadores (Arduino).
- Tener un funcionamiento automático.
- Poder detener el sistema mientras esté en funcionamiento.

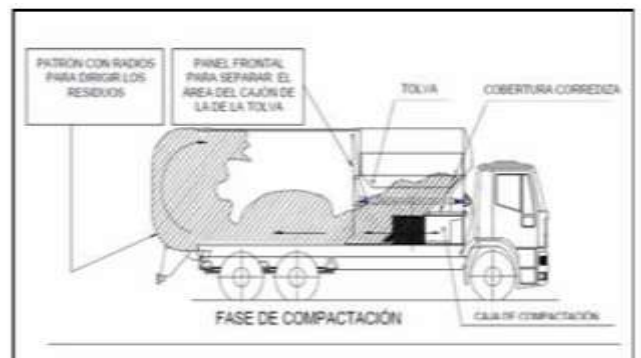


Figura 6: Esquema de funcionamiento de la caja recolectora.

La Figura 6, se muestra la estructura del funcionamiento de la caja recolectora tradicional, se muestra para tener

encueta para posteriores modificaciones y adaptaciones a nuestro proyecto, es decir lo tomamos de ejemplo para poder construir nuestro prototipo.

### Programación

En la programación se controlará los sensores del camión de basura los cuales servirán para recolectar información física del entorno y poder declararlas a nuestra comodidad. También se controlaran los motores incluyendo el inversor de giro con sus respectivos pistones a través del módulo puente H de arduino utilizando la librería stepper.h [5] para controlar la velocidad, se accionará el step up para el correcto control del motor, el inversor de giro está programado para cuando el camión no detecte en una primera instancia algún tacho de este modo el programa está diseñado para que cuando ocurra esto el camión retroceda utilizando el control de cambio de giro de los motores.

```

const int Trigger = 11; //Pin Digital 1 para el Trigger del sensor
const int Echo = 3; //Pin digital 3 para el Echo del sensor
//Declaración
#define AIN1 A1 // pin 1 de dirección del Motor Izquierdo
#define AIN2 A2 // pin 2 de dirección del Motor Izquierdo
#define PWM1 6 // pin PWM del Motor Izquierdo
#define BIN1 A3 // pin 1 de dirección del Motor Derecho
#define BIN2 A4 // pin 2 de dirección del Motor Derecho
#define PWM2 5 // pin PWM del Motor Derecho
#define STEP 20 // Control de Posicionamiento de los motores General

#define Sensor1 7 // pin 1 de dirección del Motor Derecho
#define Sensor2 4 // pin PWM del Motor Derecho
#define Sensor3 2 // Control de Posicionamiento de los motores General
int SEN_IZQUIERDA, SEN_MEDIO, SEN_DERECHA;
//parte

#include <Stepper.h>

Stepper arm( // Servo de elevación
Servo aqa; // Servo de apazte

void setup() {
  //Inicialización
  Serial.begin(9600); //inicializame la comunicación
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW); //inicializame el pin con 0

  // put your setup code here, to run once:
  pinMode(AIN1, OUTPUT);
  pinMode(AIN2, OUTPUT);
  pinMode(PWM1, OUTPUT);
}

void angulomenos(int angle,int secs){
  for (int i=0;i<angle;i++){
    arm.write(angle-i);
    delay(secs);
  }
}

void ctriangulomas(int angle){
  angulomas(angle,10);
}

void ctriangulomenos(int angle){
  angulomenos(angle,10);
}

}

void recoger(){
  // Bajar Brazo
  ctriangulomenos(130);
  delay(200);
  // Agarrar Cubo
  aqa.write(90);
  delay(200);
  // Levantar Brazo
  ctriangulomas(130);
  //delay(200);
  // Borrar Cubo
  aqa.write(160);
}

void descanso(){
  Serial.print("Posición Inicial de la Garra");
  arm.write(130);
  aqa.write(130);
}

```

Figura 6: Programación en Arduino

### Cálculo del brazo robótico

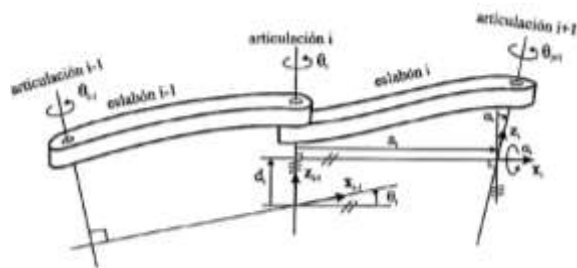


Figura 7: Cinemática Directa

$\theta_i$  Es el ángulo que forma los ejes  $x_{i-1}$  y  $x_i$  medido en un plano perpendicular al eje  $z_{i-1}$  usando la regla de la mano derecha. Se trata de un parametro variable articulado.

$d_i$  Es la distancia a lo largo de eje  $z_{i-1}$  desde el origen del sistema de coordenadas (i-1)-ésimo hasta la intersección del eje  $z_{i-1}$  con el eje  $x_i$ .

$a_i$  Es la distancia a lo largo del eje  $x_i$  que va desde la intersección de eje  $z_{i-1}$  con el eje  $x_i$  hasta el origen del sistema i-ésimo, en el caso de articulaciones giratorias.

$\alpha_i$  Es el ángulo de separación del eje  $z_{i-1}$  y el eje  $z_i$ , medido en un plano perpendicular al eje  $x_i$ , utilizando la regla de la mano derecha.

### CALCULO DE LA MATRIZ HOMOGENEA

Dada la matriz de transformación homogénea de la siguiente forma:

$${}^{i-1}A_i = T(z, \theta_i)T(0, 0, d_i)T(a_i, 0, 0)T(x, \alpha_i)$$

Realizando el producto entre las matrices. ( $\theta_i$ ,  $a_i$ ,  $d_i$ ,  $\alpha_i$  son los parámetros D-H del eslabon i).

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 & 1 & 0 & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obteniendo como resultado

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### ALGORITMO DE DENAVIT-HARTENBERG

Este algoritmo nos permite hallar la cinemática directa de un robot relacionando los sistemas de coordenadas de un eslabón y el siguiente mediante un par de traslaciones y rotaciones simples. Los pasos del algoritmo de Denavit-Hartenberg (D-H) son 16.

Eslabon	Longitud (cm)	Rango (°)
1 (L1)	2	0 a 180°
2 (L2)	4	0 a 180°
3 (L3)	8	0 a 180°
4 (L4)	8	0 a 180°
PINZA	7	0 a 45°

Tabla 1: Características de eslabones y articulaciones



## MODELO CINEMATICO DIRECTO

La cinemática directa consiste en determinar la posición y orientación del extremo final del robot con respecto al sistema de la base del robot a partir del conocimiento de los valores de las articulaciones y sus parámetros geométricos [6].

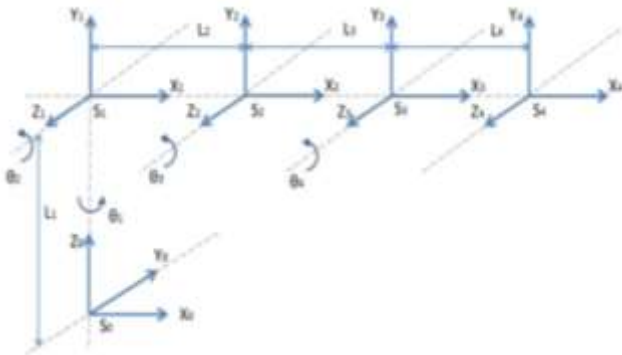


Figura 8: Representación del robot en sistemas de coordenadas según algoritmo de D-H

Obtenemos los parámetros de nuestro robot:

ARTICULACION	$\theta$	d	a	$\alpha$
1	$\theta_1$	L1	0	90
2	$\theta_2$	0	L2	0
3	$\theta_3$	0	L3	0
4	$\theta_4$	0	L4	0

Tabla 2: Parámetros de D-H

A partir de estos parámetros se obtuvieron las ecuaciones que permiten calcular los valores de la posición del extremo del robot en coordenadas cartesianas.

$$X_0 = 4 * C1 * C2 + 8 * C4 * (C1 * C2 * C3 - C1 * S2 * S3) - 8 * S4 * (C1 * C2 * S3 + C1 * C3 * S2) + 8 * C1 * C2 * C3 - 8 * C1 * S2 * S3 \dots (i)$$

$$Y_0 = 4 * C2 * S1 - 8 * C4 * (S1 * S2 * S3 - C2 * C3 * S1) - 8 * S4 * (C2 * S1 * S3 + C3 * S1 * S2) - 8 * S1 * S2 * S3 + 8 * C2 * C3 * S1 \dots (ii)$$

$$Z_0 = 4 * S2 + 8 * C2 * S3 + 8 * C3 * S2 + 8 * C4 * (C2 * S3 + C3 * S2) + 8 * S4 * (C2 * C3 - S2 * S3) + 2 \dots (iii)$$

## MODELO CINEMATICO INDIRECTO

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot, para que su extremo se posicione y oriente según una determinada localización espacial. La resolución no es sistemática, depende de la configuración del robot y pueden existir soluciones múltiples [7].

Para robots de pocos grados de libertad, como es el caso del manipulador del robot, es conveniente usar métodos geométricos (Fig. 9) los cuales utilizan relaciones trigonométricas y geométricas en los elementos que se forman en el robot.

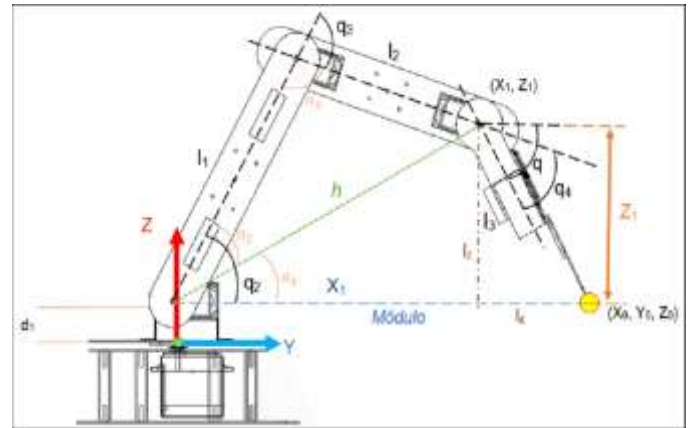


Figura 9: Representación del manipulador para obtener el modelo cinemático inverso mediante el método geométrico.

Las ecuaciones obtenidas a partir de este método son las siguientes:

$$q_1 = \tan^{-1} \left( \frac{Y_0}{X_0} \right) \dots (iv)$$

$$lx = l_4 * \cos(q) \dots (v)$$

$$lz = l_4 * \sin(q) \dots (vi)$$

$$X_1 = \sqrt{X_0^2 + Y_0^2} - lx \dots (vii)$$

$$Z_1 = Z_0 - lz * (d_1 + l_1) \dots (viii)$$

$$h = \sqrt{X_1^2 + Z_1^2} \dots (ix)$$

$$q_2 = \tan^{-1} \left( \frac{Z_1}{X_1} \right) + \cos^{-1} \left( \frac{l_2^2 - l_3^2 + h^2}{2 * l_2 * h} \right) \dots (x)$$

$$q_3 = \cos^{-1} \left( \frac{l_2^2 - l_3^2 - h^2}{2 * l_2 * h} \right) - 180^\circ \dots (xi)$$

$$q_4 = q - q_2 - q_3 \dots (xii)$$

Las ecuaciones (iv a la xii) permiten calcular los ángulos de rotación de las articulaciones

Donde:

$q_1, q_2, q_3, q_4$ : posiciones articulares.

$x_0, y_0, z_0$ : posición deseada para el extremo

## CONTROL CINEMÁTICO Y DINÁMICO DEL ROBOT

### Generación de trayectorias

Para realizar una tarea determinada el robot debe moverse desde un punto inicial a uno final. Este movimiento puede ser realizado según infinitas trayectorias espaciales. De todas ellas hay algunas que, bien por su sencillez de implementación por parte del control cinemático o bien por su utilidad y aplicación a diversas tareas, son las que en la práctica incorporan los robots comerciales.

### Trayectorias Punto a Punto

En este tipo de trayectorias cada articulación evoluciona desde su posición inicial a la final sin realizar consideración alguna sobre el estado o evolución de las demás articulaciones. Normalmente, cada actuador trata de llevar a su articulación al punto de destino en el menor tiempo posible, pudiéndose distinguir dos casos: movimiento eje a eje y movimiento simultáneo de ejes.

### Trayectorias Coordinadas o Isocronas

Para evitar que algunos actuadores trabajen forzando sus velocidades y aceleraciones, teniendo que

esperar después la conclusión del movimiento de la articulación más lenta, puede hacerse un cálculo previo, averiguando cuál es esta articulación y qué tiempo invertirá. Se ralentizará entonces el movimiento del resto de los ejes para que inviertan el mismo tiempo en su movimiento, acabando todos ellos simultáneamente. Se tiene así que todas las articulaciones se coordinan comenzando y acabando su movimiento a la vez, adaptándose todas a la más lenta. El tiempo total invertido en el movimiento es el menor posible y no se piden aceleraciones y velocidades elevadas a los actuadores de manera inútil.

### Trayectorias Continuas

Cuando se pretende que la trayectoria que siga el extremo del robot sea conocida por el usuario (trayectoria en el espacio cartesiano o de la tarea), es preciso calcular de manera continua las trayectorias articulares.

Para el manipulador del robot se implementaron todos los tipos de trayectorias mencionados anteriormente, vale destacar que para el caso de la trayectoria continua se diseñó un generador de trayectoria lineal basado en la ecuación paramétrica de una recta en tres dimensiones (xiii), donde:

**Pi:** punto inicial

**Pf:** punto final

**n:** cantidad de puntos intermedios deseados por el usuario.

La variación del parámetro (t) permite hallar todos los puntos intermedios entre el punto inicial y el final, el rango de valores posibles es:  $(0 \leq (t) \leq 1)$ , coincidiendo los extremos con los puntos inicial y final respectivamente.

Finalmente el sistema de control del manipulador ejecuta cada posición entregada por el modelo de forma secuencial asegurando que la trayectoria descrita por el extremo sea la deseada. Mientras mayor sea el valor de (n) la trayectoria resulta más precisa.

$$Px = P_i(x) + (P_f(x) - P_i(x)) * t$$

$$Py = P_i(y) + (P_f(y) - P_i(y)) * t$$

$$Pz = P_i(z) + (P_f(z) - P_i(z)) * t \dots (xiii)$$

### Control Dinámico

El control dinámico tiene como objetivo procurar que las trayectorias realmente seguidas por el robot sean lo más parecidas posible a las propuestas por el control cinemático como trayectorias deseadas. Para llevar a cabo esa tarea hace uso del conocimiento del modelo dinámico del robot y de las herramientas de análisis y diseño aportadas por la teoría de control (estabilidad de Lyapunov, control PID, control adaptativo, etc.).

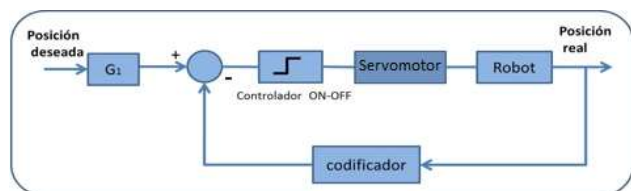


Figura 10: Esquema del control de posición para una articulación del manipulador

### III. CONCLUSIONES

El algoritmo implementado permite la programación de tareas en los espacios articular y cartesiano, así como aplicaciones de manipulación de objetos en las cuales se

conozca la posición del mismo en el espacio de trabajo, realizando trayectorias punto a punto, coordinado y lineal en el extremo. Se pretende continuar la investigación mediante la incorporación de sensores externos que permitan ampliar las esferas de aplicaciones, además de la incorporación de una cámara fija para la integración robot-visión en el espacio de trabajo.

### IV. OBSERVACIONES

Los experimentos realizados a través de la interfaz y el manipulador permitieron validar los modelos cinemáticos directo e inverso, posicionando el extremo del manipulador en los valores fijados por el usuario, esto se comprobó de manera práctica utilizando instrumentos de medición.

Se establecieron varias posiciones para el extremo y para cada articulación, y siempre la diferencia entre los valores reales a los que llegaba el manipulador y los deseados por el usuario eran menos de 1 cm, o sea, que el método desarrollado fue bastante preciso para este tipo de aplicación.

### V. REFERENCIAS

- [1]. NAVARRO MACÍAS, R: Como los sistemas de basura inteligentes reducen los costes de recogida, Recuperado de: www.lavanguardia.com, 2017.
- [2]. GARCÍA D.D, MANUEL: "Recogida de residuos urbanos", Director de Operaciones de LIMPASAM, Sevilla, 2007.
- [3]. ANA, G; LOPEZ, M. Y ORTIZ, L.: "Ideas para gestion de residuos eficiente. Diseño de un contenedor para aluminio", Arquitectura sostenible, medioambiente y eficiencia energetica 18 (2): 71-72, 2011.
- [4]. FUENTES, R. y JUAREZ, C.: Clasificación Automático de Residuos Sólidos (CARS), Ed. Universidad Nacional Autonoma de Mexico, 2013.
- [5]. SNYDER, B. J.; J. BUSSARD; J. DOLAK y T. WEISER: "A Portable Sisal Decorticator for Kenyan Farmers", International Journal for Service Learning in Engineering, 2 (1): 92-116, 2006.
- [6]. VERA J. L.: "Control de trayectorias de la simulacion de un brazo de 5 grados de libertad, controlado mediante la plataforma C2000 Piccolo LAUNCHXL-F28", Applied Engineering in Agriculture, 20 (2): 139-145, 2017.
- [7]. VERA J. L.: "Control de trayectorias de la simulacion de un brazo de 5 grados de libertad, controlado mediante la plataforma C2000 Piccolo LAUNCHXL-F28", Applied Engineering in Agriculture, 20 (2): 145-150, 2017.

### VI. BIOGRAFIAS



**RALF VIVANCO**

Nació en el departamento de Lima el 5 de marzo del 1994.

Universidad Ricardo Palma- Perú. Actualmente es estudiante del 8vo ciclo de Ingeniería Mecatrónica.



**GERSON COLLANTES**

Nació en el departamento de Lima el 25 de diciembre del 1995.

Universidad Ricardo Palma- Perú. Actualmente es estudiante del 8vo ciclo de Ingeniería Mecatrónica.



**ABEL ARBILDO**

Nació en el departamento de Lima el 23 de Julio de 1995.

Universidad Ricardo Palma- Perú. Actualmente es estudiante del 8vo ciclo de Ingeniería Mecatrónica.