Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

**Article Citation Format**
A. Oloyede, D. Ogunlana, P. Akinyemi, A.P. Oduroye & O.B. Longe (2017). Deductive Review: Dissimilarities and Similarities of Wired and Wireless Computing Networks. Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 5, No. 2. Pp 175-194

**Article Progress Time Stamps**

**Article Type:** Research Article
**Manuscript Received:** 8rd March, 2017
**Review Type:** Blind
**Review/Acceptance Information Sent** : 10th May, 2016
**Final Acceptance:**: 14th May, 2017
**DOI Prefix:** 10.22624

# Development of a Mobile Controlled Robot Using Artificial Vision with Advanced Intelligent System

**A. Oloyede, P. Akinyemi, A.P. Oduroye & O.B. Longe**
Department of Computer Science
Caleb University Imota Lagos, Nigeria.

**D. Ogunlana**
Department of Computer Science
National Open University of Nigeria
Abuja, Nigeria.

Corresponding Author Email: ayglo55@yahoo.com
Phone Number: +2348033937115

## ABSTRACT

This developed system of mobile robot is controlled by means of artificial vision, capable of recognizing, grabbing and moving specific objects in a completely autonomous way is presented, together with the conceptual and theoretical-practical grounds for the work. A mechanically robust robot is built and a system is designed, allowing the mobility of two sensors jointly, i.e., artificial vision camera and distance sensor. This makes it possible to improve the range of artificial vision, over approximately 180°, achieving precise positioning of the mobile robot. The artificial vision camera, CMUCam 2, provides the mobile robot with great autonomy which helps in interaction with its surrounding world. Having a mobile robot like this will allow interesting developments to be made in various areas of mobile robotics.

**Keywords:** Mobile Controlled Robot, Artificial Vision, Advanced Intelligent System & Camera

## 1. INTRODUCTION

In recent years, there has been increasing interest in research on mobile robots due to the endless number of remote applications that can be developed with them, particularly in areas of high risk to human beings. Currently, thanks to the degree of development that has been reached in the field of mechanisms and sensors, the implementation of systems with a high degree of freedom. This has allowed robots to get a more accurate description of their environment, such as sampling, analysis of the surrounding, detection of gases leakage, or even sending video signals so that the observer is not exposed. Since mobile robots can now carry out their tasks with greater accuracy, precision and autonomy, the developments of control methods, as well as the implementation of new prototypes, are of high importance in the field of robotics.[1]

Mobile robots exhibit the advantage of integrating the functionality of a device or system with the mobility of an autonomous vehicle, offering a number of advantages, but also a number of yet unsolved technological barriers such as:

- Increased duration of battery charge to achieve greater autonomy.
- Increased potentialities of the navigation systems to allow automatic mobility in the most efficient, flexible, fault-tolerant and safe possible manner.
- Improved efficiency of the control of fleets consisting of various mobile robots to solve problems such as optimized scaling, routing or traffic management, etc.

## 2.0 RELATED WORK

Location is one of the key technologies in autonomous mobile robot navigation, as it is the foundation of the route plan and obstacle avoidance of mobile robots. The work of [10] is concerned with the problem of determining the position of mobile robots by vision. A type of infrared landmark was designed; a system software based on Visual C++6.0 and OpenCV was build; and a location system for vision-based mobile robot and artificial landmark was developed. The infrared landmark image was acquired by a vision sensor and its mass center in the image was recognized by image processing. By the triangulation method, the robot's position in the world coordinate system was obtained. Experimental results show that the method could be applied in the field of self-localization of mobile robots [2]

On the other hand, this study focuses not only on controlling a mobile robot through artificial vision, but also on controlling the vision of one camera by the joint work of algorithms and the robot's motion. In [7] a vision tracking system for mobile robots using Unscented Kalman Filter (UKF) was introduced. The proposed system accurately estimates the position and orientation of the mobile robot by integrating information received from encoders, inertial sensors, and active beacons. These position and orientation estimates are used to rotate the camera towards the target during robot motion. The UKF, employed as an efficient sensor fusion algorithm, is an advanced filtering technique that reduces the position and orientation errors of the sensors.

Recent research by [3][4] present and emulate a wheeled robot for object transportation in a distribution center. The robot follows a free trajectory, which is controlled by artificial vision and fuzzy logic modules. The artificial vision system includes a webcam located in the upper part of the distribution center, which was used to calculate the location of the robot. Specifically, image segmentation technique of red color was implemented in the artificial vision system to determine the robot's position and orientation.

The information obtained from the webcam is also employed by the fuzzy controller to estimate the robot's velocity, which is then send to the mobile robot wirelessly. The control of the motors and the wireless communication of the robot are performed by an Arduino platform, which supports an Xbee module for communications. The image processing and fuzzy control are implemented on a PC using Matlab. The light source selected generated a uniform diffuse lighting with not much glare in the environment, which enabled the artificial vision system to retrieve traits of interest from the physical surroundings. The work of [4] presents the last developments in vision-based target tracking by an Autonomous Underwater Vehicle (AUV). The main concepts behind the visual relative localization are provided and the results from a statistical analysis for the relative localization algorithm are presented. The purpose of this analysis is to ensure properness of data used to feed controllers that are responsible for governing the AUV motion. A new set of controllers enabling the AUV to track a visual target is given. Experimental data obtained from tests in tank are presented, validating both the visual relative localization and control of the AUV[5]

## 2.1 General Aspects of Robotics

### 2.1.1 Background of Mobile Robots

The use of mobile robots is justified in applications in which tasks that are unpleasant or risky to humans are carried out, such as transporting hazardous materials, mining excavations, industrial cleaning, or inspection of nuclear power plants. In these situations, a mobile robot can perform the task and avoid uncalled-for risks to the workers' health. Another group of applications in which these kinds of robots complement the operator's activities are supervision, inspection, or assisting disabled people. Also, in surgical teleoperation applications, where there is considerable backwardness in communications, the use of vehicles with some degree of autonomy is interesting [6]

An autonomous mobile robot is characterized by having an intelligent connection between the operations of perception and action, which defines its behavior and allows it to fulfill the programmed objectives in surroundings with some uncertainty. The degree of autonomy depends largely on the ability of a robot to disregard the surroundings and convert the obtained information into orders, so that when orders are applied to the locomotion system's actuators, the efficient performance of the task is guaranteed. From this, it can be concluded that the main characteristics of a mobile robot, in contrast with any other kind of vehicle, are the following:

- Perception, which determines the relation of the robot with its work environment, by means of its onboard sensors.
- Reasoning, which determines the actions that must be carried out at all times according to the state of the robot and its surroundings to reach the assigned goals.

In this way, the reasoning ability of an autonomous mobile robot is translated into the planning of safe trajectories that will allow it to fulfill the assigned objectives. The execution of a task in particular must be carried out in a closed loop for the robot to adapt to navigating through unstructured surroundings. A traditional control loop is not used because the action is not generated by simple output feedback. It is therefore necessary to use a planner skilled in geometric analysis that knows the conditions of the surroundings and of the mobile robot, together with its kinematic and dynamic characteristics. Starting from this point, the mobile robot can transform data supplied by the perception into adequate control references that do not go beyond any of its physical limitations, and that define trajectories free from obstacles that guarantee the achievement of the goals established for the specific task. Thus, the planner becomes responsible, to a large extent, of the mobile robot's navigation efficiency; therefore its design requires special care [8] [9]

Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

To carry out these tasks it is necessary to have fine control of the movements of the mobile robot. The following techniques stand out for this purpose:

- Pure Pursue.
- PI-PD-PID control techniques.
- Ziegler-Nichols controllers.
- Cohen and Coon methods.
- Fuzzy control.

In addition to the above control techniques, there are methods for the execution of its specific tasks, e.g., the geometric method for following routes explicitly, the methods based on control theory applied to following trajectories and routes, as well as predictive and reactive control following, based on the direct reaction to environmental perception sensors [2].

### 2.1.2 Morphology of Mobile Robots

The design of the mobile robot that will be constructed is based on a configuration that consists of four wheels. Various configurations that comply with these characteristics are presented below ([9] [5] [10] [8]

#### 2.1.2.1 Ackermann

The arrangement and mobility of its axles is similar to that of a conventional vehicle. The inner front wheel turns at a slightly larger angle than the outer wheel to prevent sliding. The extensions of the front axles intersect at some point on the extension of the axle of the rear wheels. The locus of the points drawn on the ground by the centers of the wheels is concentric circumferences centered on axle P1, as shown in Figure 1.

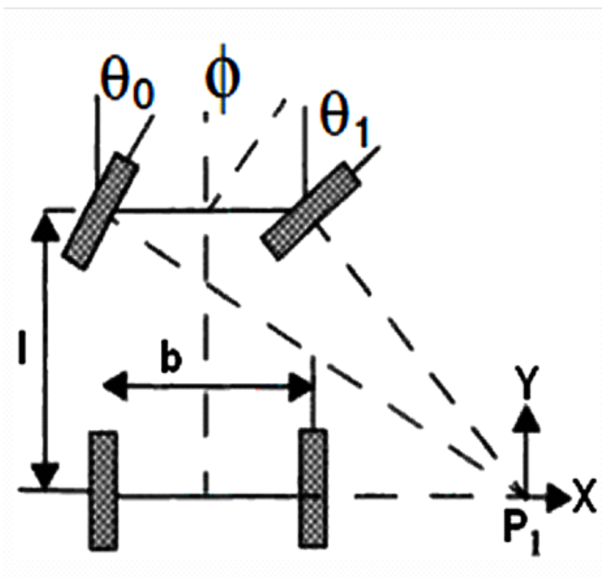


Figure 1. Ackermann configuration

#### 2.1.2.2 Skid-Steer

They are composed of several wheels on both sides of the vehicle that act simultaneously. The motion consists of the combination of the movements of the left and right sides, which result in advance, reverse, turning motions, etc. It is important to point out that, in contrast with the Ackermann, there is no axle movement. Other kinds of four-wheel robots exist with hybrid configurations of wheels with differential traction and omnidirectional, as well as others with four omnidirectional wheels. For the design of the mobile robot presented in this chapter, this four-wheel configuration without axle movement was chosen, thanks to its great maneuverability.

## 3. KINEMATICS OF MOBILE ROBOTS

Kinematics is the simplest study of the behavior of a mechanical system. In mobile robots, it is necessary to understand the mechanical behavior of both the approximate design for the task to be performed and the process of creation of the control software for the robot's hardware. Mobile robots are not the first complex mechanical systems to require such analysis. Manipulator robots have been subjected to intensive research for over 40 years (from the 70s), because to a certain extent they are more complex than other mobile robots, thus the robotics community has reached a deep understanding of the kinematics and even of the dynamics of manipulator robots.

### 3.1 Representation of the Robot's Position

In the kinematic analysis, the mobile robot is considered as a rigid body on four wheels operating on a horizontal plane. To specify the position and orientation of this mobile robot, a two-dimensional movement is established between a global reference framework of the ($X_g$, $Y_g$, $Z_g$) plane that is chosen arbitrarily, and a local reference framework of the robot ($x_1$, $y_1$, $z_l$), associated with a point on the frame that corresponds to the robot's Center of Mass (COM). According to Figure 2, this COM can be described in the global reference system as COM=(X, Y), and the angular difference between the global and local system of reference is θ. Therefore, its position and orientation are determined by equation (1):

$$\varepsilon_c = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} \qquad (3.1)$$

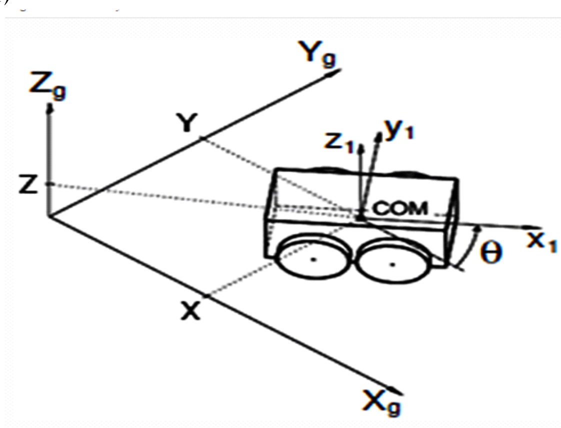


Figure 2. Velocities of the robot and global reference

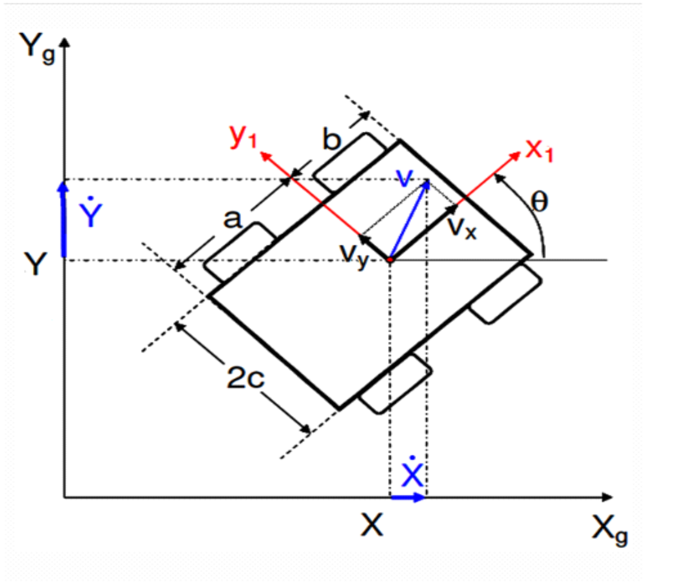


**Figure 3: Reference System**

The velocity vector is defined on a two-dimensional coordinate system of the COM and expressed as a function of the local reference system by means of $V_x$ and $V_y$, which determine the vehicle's longitudinal and lateral velocities (Figure 3). To obtain the robot's movement in global terms, its local movement is mapped along the global reference axes, leading to equation (2):

$$q = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} = \begin{pmatrix} \cos\theta - & sen\theta & 0 \\ sen\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} V_Z \\ V_y \\ w \end{bmatrix} \quad (3.2)$$

### 3.2 Velocity Ratio

The representation of the mobile robot in space does not impose any restriction on its motion in the plane, therefore only its free body kinematics is described. However, it is necessary to include the ratio between the velocities of the wheels and the local velocities. Then we have that the cinematic model for this mobile robot is constituted by the system of equations (3), (4), (5):

$$\dot{X} = r\left(\frac{w_t + w_R}{2}\right)\cos\theta + x_{ICR} r\left(\frac{-w_l + w_R}{2c}\right) sen\theta \quad (3.3)$$

$$\dot{Y} = r\left(\frac{w_l + w_R}{2}\right) sen\theta + x_{ICR} r\left(\frac{-w_l + w_R}{2c}\right)\cos\theta \quad (3.4)$$

$$\dot{\theta} = r\left(\frac{-w_l + w_R}{2c}\right) \quad (3.5)$$

Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

where $x_{ICR}$ is a coordinate of the center of instantaneous rotation, and $w_l$ and $w_R$ denote the angular velocities of the left and right wheels, respectively. At the kinematic level, they can be considered as control entries and can be used to control the longitudinal and angular velocity according to equation (6): $x_{ICR}$

$$\omega = r\left(\frac{-w_l + w_R}{2c}\right) \tag{3.6}$$

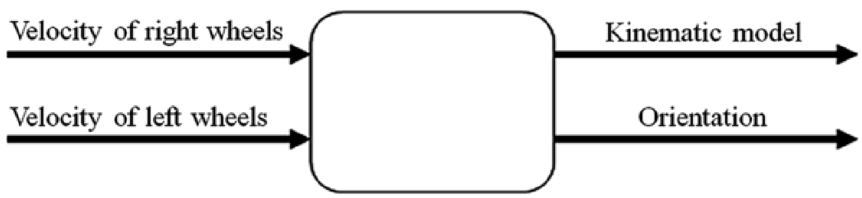


**Figure 4. Kinematic control of the robot**

### 3.3 Guidance of Mobile Robots

The navigation problem of a mobile robot can be summarized in three questions: Where am I? Where am I going? And how should I get there? therefore a navigation system consists in the set of sensors, systems, methods and technologies that try to situate a mobile robot in its surroundings. Now, for a mobile robot to be completely autonomous and be able to go over its workspace without the occurrence of any inconveniences, it is necessary to answer the above questions by means of the following stages in the process of mobile robot guidance: localization, mapping and planning of trajectories.

The localization process consists in finding the location of the mobile robot within its work setting. The mapping process consists in creating a representation of the setting in which the mobile robot is performing, which must be comprehensible by it. The trajectory planning consists in determining the route that the mobile robot must follow, from an initial to a final configuration. This trajectory must be planned in such a way that the mobile robot does not collide with objects that it finds along its way.

#### 3.3.1 Localization

The concept of localization arises as a need to estimate the position of the mobile robot within the work setting (map). There are different localization techniques that can be grouped into two kinds:

- Local Localization, which consists in determining the position of the mobile robot from a known initial position; *i.e.*, follow-up the movements made by the mobile robot, estimate its degree of advance and orientation, and make calculations to estimate the current position from the starting point of the movement, for example, odometry and landmarks may be mentioned.
- Global Localization, which consists in finding the position of the mobile robot without knowing its location at a previous instant, or having uncertain information. Kalman filtering, Markov filtering, and particle filtering may be mentioned.

### 3.3.2 Map Generation

It is important for the mobile robot to get in some way a map of the surroundings in which it must move. This must be done so that in a later stage it can use that map to trace the trajectory that it must follow. In general, maps are not adapted to the requirements of the robot and elaborating them manually is a time-consuming and costly process. If a robot builds its own map, the map will be suitable for the robot sensory capacities, and thus for use in navigation or localization.

- **Sensory Maps:** They are based on invariant characteristics or on grid maps of spatial occupation (probabilistic). In the grid maps the space is represented as a set of 2D or 3D cells, associating an occupation (1) or free (0) state. The number of cells is ($n^d$), where $n$ is the number of cells per dimension and $d$ is the number of dimensions.
- **Geometric Maps:** It is considered that the space can be represented by a series of geometric primitives (such as straight lines or polygons). The exploration is based on algorithms inspired in computational geometry. For instance, there are some techniques based on reaching a goal by moving in a straight line.
- **Topological Maps:** A map of the different situations that the robot will face is constructed. Then this map is represented by means of a graph, so when the map indicates a change of situation, i.e., differences with the expected, a new node and an arc are included. The resultant graph represents a map of the free space of the surroundings.

### 3.3.3 Trajectory Planning

The trajectory-planning problem consists in determining a trajectory between an initial and a final configuration, so that the mobile robot does not collide with the obstacles and complies with its kinematic restrictions. To do this it is necessary to consider the following specifications: minimum distance trajectory, semi-unknown environments, dynamic environments, additional restrictions, and efficiency. To tackle the trajectory planning problem there are various methods that are grouped into graph search, dynamic programming, Voronoi diagrams, and visibility graphs.

### 3.4 Choice of Electric Components

Each of these elements must be chosen according to the task for which the robot was constructed, since they play a fundamental role for the configuration of the robot to perform specific tasks. It is necessary to know in advance what these specific tasks are in order to determine the electrical elements to be used.
The electric components chosen for the designed and implemented mobile robot are presented below:

- **Servomotors:** The total number of servomotors of the mobile robot is 17.
- **Microcontroller:** The microcontroller used is a Basic Stamp by Parallax.
- **Servo controller:** Because of the large number of servomotors used in the configuration of the mobile robot and for a better compatibility performance, the Parallax Servo Controller (PSC) board is used.
- **Sensors:** Due to the various tasks that this mobile robot will carry out, two types of sensors are chosen: an ultrasonic distance sensor, and a shape and color sensor (Acroname CMUCam 2+).
- **Batteries:** Two 6-V rechargeable batteries are used.
- **Computer:** To improve the processing capacity with respect to the artificial vision task, a netbook was incorporated.

### 3.5 Construction of the Mobile Robot

The construction process of this mobile robot was divided into three main mechanical stages: the frame of the mobile robot, a manipulator with clamp as terminal effector, and a manipulator of the artificial vision camera.

### 3.5.1 Frame of the Mobile Robot

The frame of the mobile robot is the structure that integrates all its components, *i.e.*, the camera with its manipulator, the manipulator, batteries, circuits, and sensors, among other components, forming the robot's main structure. The design criteria were selected considering that the mobile robot has the following characteristics:

- Great mobility,
- Acceptable autonomy time,
- Light and easily transportable,
- Expansion and connectivity capacity,
- Acceptable construction cost.

A rail capable of displacing the artificial vision camera linearly to increase its range of vision is designed. This configuration was performed imitating the motion of the head of a turtle, in this way increasing its vision range by almost 180°.

Therefore, in its final design, the mobile robot consists of the following elements:

- **Base0:** Inside there are two supports for each wall for the servomotors in charge of the traction.
- **Base1:** This base contains all the robot's wiring and the battery supports, as well as the converters that allow communication between the camera and the netbook
- **PC Casing:** It is mounted on top of Base1, and its main function is to protect the netbook from possible impacts.
- **Manipulator Support:** It is fixed on Base1 and rests on the PC casing to provide more stability and strength to the manipulator.
- **Microcontroller Casing:** Its function is only to protect the microcontroller and its servo controlling board from dust as well as from possible impacts, in addition to avoiding possible accidental disconnections of elements like the sensors, servomotors, artificial vision camera, and batteries.

Figure 5 shows the above-mentioned elements, assembled to form the frame.

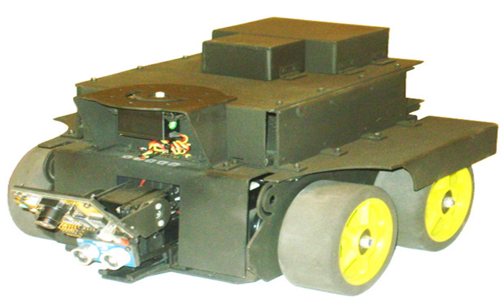

**Figure 5. Frame of the mobile robot**

### 3.5.2 Manipulator

This manipulator has four all rotational DOF on the same axle, and a terminal effector composed of a clamp. Out of these DOF, the first one allows a 180° turning motion that enables the handling and deposit of grasped objects. The second one allows the reaching motion and practically has a function akin to that of the shoulder joint. The third allows bringing close, fulfilling the function of the elbow, also with a 180° motion; and the fourth allows a fine and precise motion, serving a function similar to the motion of the human wrist, with a movement over 180°. The terminal effector allows grasping objects, and in general performs the function of the human fingers.

Figure 6 illustrates the manipulator, which is composed of the following mechanical elements: base manipulator, arm, forearm, wrist, and terminal effector.

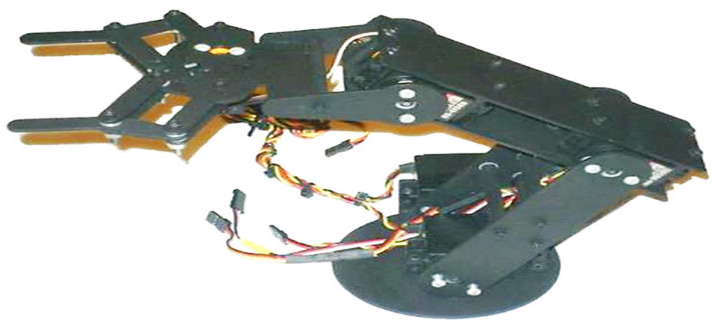

**Figure 6. Manipulator arm**

### 3.5.3 Camera Manipulator

The camera manipulator consists of the following elements:

- **Camera-Sensor Support:** It supports the camera as well as the ultrasonic sensor, and it also performs the PAN motion by means of the servomotor.
- **TILT Support:** The part supports the servomotor in charge of performing the PAN movement.
- **Cart:** It has the function of sliding the artificial vision camera into or out of the frame of the mobile robot, to expand its vision range.
- **Adjustment Support:** It consists of two parts that are fixed on base0, whose function is to provide mechanical adjustment to the cart.

These three mechanical parts, assembled, constitute the mobile robot shown in Figure 7.

Digital Innovations & Contemporary Research in SCIENCE & ENGINEERING
Vol. 5 No. 2, June 2017

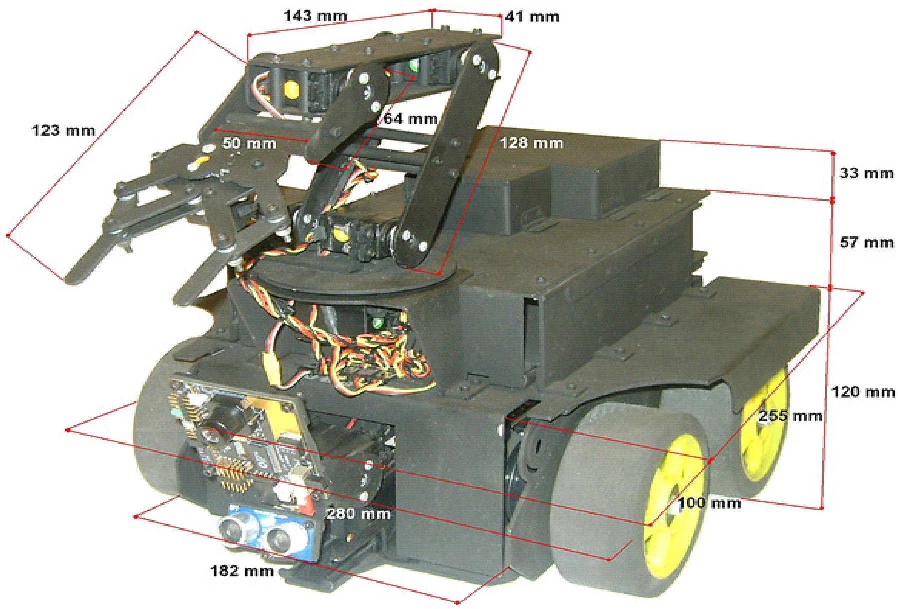


Figure 7: Designed and implemented mobile robot

## 4. PROGRAMMING AND RESULTS

The microcontroller used for programming the mobile robot is a Basic Stamp, whose programming language is Basic. On the other hand, the commands of the artificial vision camera are very important for the development of object recognition, because if one of these commands is poorly configured, the recognition cannot take place. In addition to these commands, the artificial vision camera has others with different work characteristics and properties. The advantage of using this artificial vision camera is that, by means of its own commands, it allows the use of a technique for recognizing objects by means of their color. The mobile robot also has a netbook with the purpose of making it carry out specific tasks and work in general, as complex as its programming and its sensors allow.

### 4.1 Camera Programming

The robot starts functioning by searching for an object based on its color. Thus, the first program to execute is that of the configuration of the camera, which is presented below:

```
'-----------------------------------
'CONFIGURACION '-----------------------------------
CONFIGURACION_CMU:
        SEROUT 6, 84, ["RS",13]
        PAUSE 200
        SEROUT 6, 84, ["PM 1",13]
        SERIN 7, 84, [WAIT (":")]
        PAUSE 200
        SEROUT 6, 84, ["RM 2",13]
        SEROUT 6, 84, ["OM 0 63",13]
        SEROUT 6, 84, ["NF 10",13]
        SEROUT 6, 84, ["CR 19 32 18 40",13]
        SERIN 7, 84, [WAIT (":")]
        GOSUB PARPADEO_LED
RETURN
'-----------------------------------
'OBTENCION DE COLORES DE IMAGEN
'-----------------------------------
obtencion_color:
        SEROUT 6, 84, ["RM 3",13]
        PAUSE 200
        SEROUT 6, 84, ["TC 60 240 16 17 16 16",13]
        SERIN 7, 84, [STR datorecibido\8]
RETURN
```

The first subroutine configures the camera by means of the values 6 and 84, which correspond to the output pin of the controller and to the constant defining speed of serial transmission, respectively. Through these values, communication between the camera and microcontroller is established.

Obtaining colors becomes very simple when applying this configuration, which, as may be seen, is mainly made up of a single command line, namely TC, which defines the maximum and minimum value of each RGB. However, in this case we are only interested in red.

Below it may be seen part of the principal program, by which subroutines are directed, that has direct relation with the detection of objects in front of the camera.

```
UBICACION:
GOSUB obtencion_color
IF (datorecibido(6) >= 72) THEN
'DEBUG "Detecto Color Rojo",CR
IF (datorecibido(3) < 44) AND (datorecibido(5) < 44) THEN
        'DEBUG "Objeto rojo a IZQ",CR
         GOSUB posicion
        pos = 1
        GOSUB centrar_izq
        ELSEIF (datorecibido(3) > 44)AND(datorecibido(5) > 44) THEN
                'DEBUG "Objeto rojo a DER",CR
                GOSUB posicion
```

```
                    pos = 2
                    GOSUB centrar_der
ELSEIF ((datorecibido(3) < 44)AND(datorecibido(5) > 44))THEN
          IF ((88-datorecibido(5)) > datorecibido(3))THEN
                    'DEBUG "OBJETO ROJO DELANTE HACIA LA IZQ",CR
                    GOSUB posicion
                    pos = 3
                    GOSUB centrar_izq
          ELSEIF ((88-datorecibido(5)) < datorecibido(3))THEN
                    'DEBUG "OBJETO ROJO DELANTE HACIA LA DER",CR
                    GOSUB posicion
                    pos = 4
                    GOSUB centrar_der
          ELSEIF ((88-datorecibido(5)) = datorecibido(3))THEN
                    'DEBUG "OBJETO ROJO FRENTE A ROBOT",CR
                    GOSUB detectado
          ENDIF
ENDIF
ELSE
GOSUB mover_camara
ENDIF
```

This program runs a series of conditional statements to define whether the camera has found the red object or not, as well as in which position the object is by comparing values obtained from the camera, specifically, the coordinates of the square that encloses the object, thereby determining if the object is left, right or at the center.

The program has two types of motion, to the left and to the right, each one destined to determine more accurately the object's position. The first motion, whether to the left or to the right, has the aim of approaching the camera to the object, while the second motion conducts a fine control to place the camera in front of the object.

When the camera is in front of the object due to the fine movement of the manipulator robot, motion stops and is directed to the subroutine 'detected', this allows the robot to continue with its configuration to grab the identified object.

It must be noted that when none of the statements is true, the camera motion subroutine starts and remains in this loop until the red object is localized. Once the camera has covered its view range, close to 180°, the robot turns clockwise and starts the search again.

### 4.2 Distance Sensor Programming

The program used for programming the distance sensor is shown below:

```
'-------------------------------
distancia_adelante:
        PULSOUT         10, Trigger
        PULSIN          10, 1, rawDist
        rawDist = rawDist */ Scale
        rawDist = rawDist / 2
        cm = rawDist * * RawToCm
        PAUSE 1000
        'DEBUG "cm=", DEC cm,CR
        PAUSE 1000
        IF (cm > 10)          THEN AVANZAR
        IF ((cm >= 8) AND (cm <=10)) THEN mover_manipulador
        IF (cm <= 7)          THEN RETROCEDER
```

With this program, it is known at what distance from the sensor an object is; by means of the PULSOUT command a signal is sent through the pin to which the sensor is connected, while the information is received by the same pin using the PULSIN command, storing information in a variable previously created for such end.

### 4.3 PSC Control Card Programming

In order to control all the servomotors, a PSC control card must be programmed. The following program is used for this purpose:

```
'{$STAMP BS2px}
'{$PBASIC 2.5}
Sdat          PIN 15 ' Pin para configuración serial
Baud CON 1646 ' Constante para velocidad de 2400 baud
buff          VAR Byte(3) ' Variable temporal
FindPSC: ' Encuentra y obtiene la versión
DEBUG "Encontrando PSC", CR          'Número de la PSC.
SEROUT Sdat, Baud, ["!SCVER?",CR]
SERIN Sdat, Baud, 500, FindPSC, [STR buff\3]
DEBUG "PSC ver: ", buff(0), buff(1), buff(2), CR
STOP
```

This program in independent from the general programming of the robot. However, it is crucial to carry it out; otherwise, it will be impossible for the PSC card to communicate with the microcontroller, impeding control of each servomotor.

### 4.4 Servomotor Programming

In addition to Hitec servomotors, there are two different servomotors, namely a Futaba servomotor in charge of performing the motion to push the cart of the camera—and a GWS servomotor in charge of opening and closing the clamp. Both respond to the same pulse train to be in central position. Below is presented a table (Table 1) with the values that must be sent to each servomotor for it to carry out motion.

**Table 1. Pulse train values for servomotors**

| Servo Motor | $0^0$ | $90^0$ | $180^0$ |
|---|---|---|---|
| Hitech 311HB | 250 | 750 | 1250 |
| Hitech 475HB | 250 | 750 | 1250 |
| Hitech 965MG | 250 | 750 | 1250 |
| Hitech 7955TG | 380 | 750 | 1120 |
| Futaba 3004 | 250 | 750 | 1240 |
| GWS 2BBMG | 250 | 750 | 1250 |

### 4.5 Servomotor Initial Movement

To initiate the complete configuration of the robot, each servomotor must be in initial position. This subroutine is as follows:

```
'---------------------------------------------------------------------------

'---------------------------------------------------------------------------
mov_iniservos:
pw = 950
SEROUT 15, 1646,["!SC", 0, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 800
SEROUT 15, 1646,["!SC", 1, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 750
SEROUT 15, 1646,["!SC", 2, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
SEROUT 15, 1646,["!SC", 3, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
SEROUT 15, 1646,["!SC", 4, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 1090
SEROUT 15, 1646,["!SC", 5, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 1130
SEROUT 15, 1646,["!SC", 6, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 1070
SEROUT 15, 1646,["!SC", 7, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 700
SEROUT 15, 1646,["!SC", 8, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
pw = 750
SEROUT 15, 1646,["!SC", 9, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
RETURN
```

The motion of each servomotor is determined by a variable previously created for that end, namely the *Word*-type pw variable, which varies according to the direction towards which each servomotor carries out motion.

### 4.5.1 Camera Servomotors Motion

It comprises three servomotors: the one that pushes the cart and the other two that move the camera's manipulator. In addition, its programming includes the following subroutines:

```
'------------------------------------------------------------------------
posini_camara:
        pw = 240
        SEROUT 15, 1646,["!SC", 0, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 100
        pw = 800
        SEROUT 15, 1646,["!SC", 1, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 1000
        pw = 350
        SEROUT 15, 1646,["!SC", 2, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 200
RETURN
'------------------------------------------------------------------------
'MOVER CÁMARA
'------------------------------------------------------------------------
mover_camara:
        pwa = pwa+50
        pwb = pwa
        pw = pwa
        SEROUT 15, 1646,["!SC", 2, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 1500
        IF (pwa = 1100) THEN nueva
         GOTO salir
nueva: pwa = 350
        SEROUT 15, 1646,["!SC", 2, 8, pwa.LOWBYTE, pwa.HIGHBYTE, CR]
        GOSUB GIRO
        RETURN
salir: RETURN
```

The first subroutine aforementioned is directly related to the initial position of the manipulator robot before initiating the object search process. The second subroutine is responsible for performing the Pan motion of the camera's manipulator to localize the object.

These programs were insufficient to localize the object placed just in front of the camera. Therefore, it was necessary to configure a new subroutine called 'position' that was able to do it. The 'position' subroutine allows localizing the position of the Pan servomotor when it sweeps the surroundings searching for the object. With the values obtained from 'position' subroutine, 'centrar_izq' or 'centrar_der' subroutines are entered. These routines are in charge of moving the manipulator of the camera towards those directions and of positioning the camera in front of the object by means of the conditional statements associated.

The robot traction motion is made up of the following subroutines:

```
'-----------------------------------------------------------------------
centrar_robot:
IF (((buff(1)= 2) AND (buff(2) > 229 AND buff(2) <= 256)) OR ((buff(1)= 3) AND
(buff(2) >= 0 AND buff(2) < 9)))THEN
        GOSUB DETENER
        GOSUB calc_dis
'Giro Izquierda
ELSEIF ((buff(1)= 1) AND (buff(2)<=256)) OR ((buff(1)= 2) AND (buff(2) < 230)) THEN
        pwb = pwb+40
        SEROUT 15, 1646, ["!SC", 2, 0, pwb.LOWBYTE, pwb.HIGHBYTE, CR]
        PAUSE 100
        pw=780
        SEROUT 15, 1646, ["!SC", 3, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
        SEROUT 15, 1646, ["!SC", 4, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 300
        GOSUB DETENER
        GOSUB UBICACION
'Giro Derecha
ELSEIF ((buff(1)=2) AND (buff(2)>247) OR ((buff(1)= 3) AND (buff(2) < 256)) OR
        ((buff(1)= 4) AND (buff(2) < 146))) THEN
        pwb = pwb-40
        SEROUT 15, 1646, ["!SC", 2, 0, pwb.LOWBYTE, pwb.HIGHBYTE, CR]
        PAUSE 100
        pw=720
        SEROUT 15, 1646, ["!SC", 3, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
        SEROUT 15, 1646, ["!SC", 4, 0, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 300
        GOSUB DETENER
        GOSUB UBICACIÓN
ENDIF
```

The first part of this subroutine indicates that the robot is already in front of the object. Then the robot is directed to the 'stop' subroutine and when it returns heads to the distance calculation subroutine. The following conditional statement moves the camera to the right and turns the robot to the left, to subsequently go to the 'stop' subroutine and then to the 'location' subroutine to center the camera in front of the object one more time. Meanwhile, the other statement does the same but in the opposite direction.

### 4.5.2 Manipulator Servomotors Motion

The manipulator robot motion is defined by the following subroutine:

```
'-----------------------------------------------------------------------
mover_manipulador:
        pw = 390
        SEROUT 15, 1646,["!SC", 6, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        pw = 550
        SEROUT 15, 1646,["!SC", 7, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        pw = 600
        SEROUT 15, 1646,["!SC", 8, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 2500
        pw = 945
        SEROUT 15, 1646,["!SC", 9, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 2000
        pw = 1130
        SEROUT 15, 1646,["!SC", 6, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        pw = 1070
        SEROUT 15, 1646,["!SC", 7, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        pw = 900
        SEROUT 15, 1646,["!SC", 8, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
        PAUSE 2000
        pw = 700
        SEROUT 15, 1646,["!SC", 8, 10, pw.LOWBYTE, pw.HIGHBYTE, CR]
GOSUB fin
```

For the manipulator robot to capable of grabbing the object, the only action executed is sending different values of the pw variable to each servomotor. These values are set according to the motion performed by the manipulator robot, which is a single movement. In addition, pauses in the programming allow the associated servomotor to perform its movement without making the next one, so it can grab the object adequately. When the object has been grabbed, the manipulator robot returns to its initial position, except for the clam that is holding the object.

All the subroutines aforementioned, together with other subroutines for analysis, are part of the complete program that allows the robot to identify the object, head to it and finally grab it. Each subroutine can be configured separately to observe its functioning, regardless of the general program. Results from each subroutine can be seen by the DEBUG command and can be accessed within the program in the form of remarks, since they consume great part of the memory available in the microcontroller. Some of the results are the localization of the object, which indicate when, where and at what distance it is detected. To conclude, results are satisfactory because the robot performs the tasks for which it was programmed.

# 5. DISCUSSION

## 5.1 Problems

The manual manufacturing of the majority of pieces constituting the robot, such as chassis, brackets, basis, axles, among others, is time-consuming. Unfortunately, Basic Stamp is not the optimal microcontroller for the development of a wide range of algorithms, due to its limited memory, which does not allow the new robot to perform tasks that are more complex.

## 5.2 Challenges

Although the results obtained were satisfactory, some lines can be developed to improve and optimize the current design:

- Addition of arms capable of lifting the robot to overcome obstacles of certain height. Specifically, the gear case used in this motion needs to be improved.
- Substitution of the microcontroller for one of greater capacity in terms of memory and algorithm processing.
- Development of research on trajectory planning, mobile robots' kinematics, among others, and of algorithms for artificial vision, mainly for recognition of objects or shapes, incorporating a netbook.

# 6. CONCLUSION

The design, construction and programming of a mobile robot controlled by means of artificial vision, specifically by means of the recognition of different color objects, has been presented. This mobile robot also has a notebook that provides it with a large artificial intelligence capacity, which depends mostly on the programming given. A mechanically robust robot has been built and a system has been designed that allows the mobility of two sensors jointly, *i.e.*, artificial vision camera and distance sensor. This makes it possible to improve the range of artificial vision to approximately 180°, achieving precise positioning of the mobile robot. The Basic Stamp microcontroller provides efficiency in the algorithms, allowing good coordination between all the constitutive elements of the mobile robot.

Incorporating the PSC card allows a simpler and controlled programming of each servomotor in terms of speed and position, making the navigation system more reliable. The artificial vision camera, CMUCam 2, provides the mobile robot with great autonomy thanks to its excellent interaction with its surrounding world. Having this mobile robot facilitates to carry out interesting developments in various areas of mobile robotics.

## REFERENCES

[1] Bin, M. R. (2008). Vision based autonomous robot . Malaysia: Universiti Teknologi Malaysia.
[2] Caracciolo L. De Luca A. Iannitti S. (1999). Trajectory tracking control of a four-wheel differentially driven mobile robot. In Proceedings of the 1999 IEEE Internatonal Conference on Robotics & Automation. Detroit, MI: IEEE. 10.1109/ROBOT.1999.773994
[3] Del Pilar, F., Guarnizo, J., & Benavides, G. (2014). Emulation system for a distribution center using mobile robot, controlled by artificial vision and fuzzy logic. IEEE Latin America Transactions , 12(4), 557–563. doi:10.1109/TLA.2014.6868855
[4] Figuereido, A. B., Ferreira, B. M., & Matos, A. C. (2016). *Visión-based localization and positioning of an AUV*. Paper presented at OCEANS 2016 Conference, Shanghai, China. 10.1109/OCEANSAP.2016.7485384
[5] Holz, D., Nieuwenhuisen, M., Droeschel, D., Stückler, J., Berner, A., Li, J., Klein, R., & Behnke, S. (2014). Active recognition and manipulation for mobile robot bin picking. In F. Röhrbein, G. Veiga, & C. Natale (Eds.), *Gearing up and accelerating cross-fertilization between academic and industrial robotics in Europe* (pp. 133-153). Springer International Publishing.
[6] Ollero, A. (2001). *Robótica: Manipuladores y robots móviles*. Barcelona: Marcombo S.A.
[7] Shaikh, M. M., Bahn, W., Lee, C., Kim, T., Lee, T., Kim, K., & Cho, D. (2011). *Mobile robot vision tracking using unscented kalman filter*. Paper presented at 2011 IEEE/SICE International Symposium on System Integration (SII), Kyoto, Japan. 10.1109/SII.2011.6147622
[8] Shang, W., Yang, C., Liu, Y. & Wang, J. (2016). Design on a Composite Mobile System for Exploration Robot. *Shock and Vibration*.10.1155/2016/6363071
[9] Silva, R., García, J. R., Barrientos, V. R., Molina, M. A., Hernández, V. M., & Silva, G. (2007). State-of-the-art of the movable wheels robots. Telématique: Revista Electrónica de Estudios Telemáticos , 6(3), 1–14.
[10] Wang L. Zhao G. (2010). A localization system for mobile robots based vision. In Proceedings of the 8th World Congress on Intelligent Control and Automation. Jinan, China: IEEE. 10.1109/WCICA.2010.5554159