

# *Odometry and Teleoperation Application using NI Robotics*

Birouaş Ionuț - Flaviu  
Faculty of I.M.T, Department of  
Mechatronics  
University of Oradea,  
Oradea, România  
flaviu.ionutz@gmail.com

Arnold Nilgesz  
Faculty of I.M.T, Department of  
Mechatronics  
University of Oradea,  
Oradea, România

Mihai Cornea  
Faculty of I.M.T, Department of  
Mechatronics  
University of Oradea,  
Oradea, România

**Abstract**— This document is intended to present the progress in developing a remotely operated mobile robot that can feedback information related to its position. The equipment used for the robot frame, communication and feedback technology are used are just one alternative of achieving the teleoperation and feedback tasks, and are not to be taken as the standard or the only way of achieving the objectives. The application was programmed using LabVIEW for the remote operation of the robot, for obtaining video streaming we have used two alternatives one being a standard IP camera and the other using a USB WebCam connected to a Raspberry Pi. Positioning feedback was determined from the information provided by the two motor encoders mounted on the DC motors.

## I. INTRODUCTION

The focus of this paper is the development of a mobile robot capable of being operated remotely via Ethernet or Local Area Network (LAN) and also stream back to the user the estimated position of the robot and video.

A simplified diagram of the system is shown in figure 1.

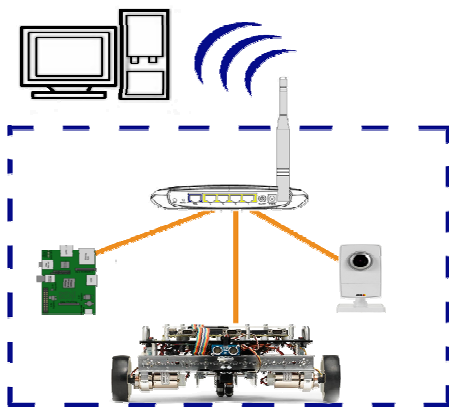


Fig. 1. Information flow diagram

## II. PRINCIPLES AND METHODS USED FOR POSITIONING FEEDBACK

Positioning estimation is achieved by using the encoders of the two DC motors. Data acquired from the encoders is processed to determine the X and Y position of the robot in a 2D environment.

Encoders on their own do not provide information such as speed of the motor or how many rotations the motor has done, in stead encoders give a pulsating signal such as TTL / Low and High. The method of generating the pulse may vary from motor to motor. The component behind most motors with feedback is the encoder wheel. Such parts are usually found directly mounted on the rotor shaft but in some cases it can be mounted on the output shaft of the gearbox. The encoder wheel is similar to a disk with small radial cutouts. An example is shown in figure 2.

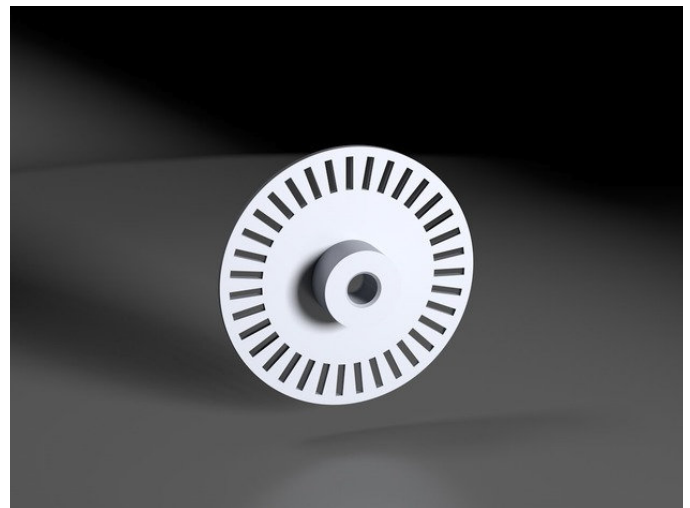


Fig. 2. Optic Encoder Wheel

This type of encoder wheels are found in optical encoders, between the 2 surfaces of the disk there are optical sensitive

sensors that act as a light barrier. Once the light barrier is interrupted between the emitter and receiver the sensor generates a logical Low or 0V. Similarly when the light barrier encoders a cutout in the encoder wheel the sensor will generate a logical High (this can be 3.3V, 5V, 10V, 12V, 24V 30V, depending on the logical circuit specifications). The principle is presented in figure 3.

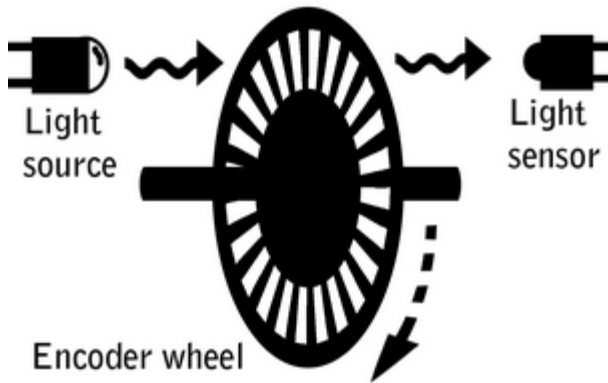


Fig. 3. Optical Encoder Working Principle

The pulse count we receive from the encoders can be integrated to determine how many rotations the motor has made and if we take into consideration the time factor we will also determine the angular velocity of the motor.

Depending on how many pulsations per rotation the encoder generates the accuracy of our determinations will vary, the more the ratio between pulsations per rotation the more accurate our determinations will be.

In figure 4 we can see an example of square wave generated by an encoder and capture using an oscilloscope.

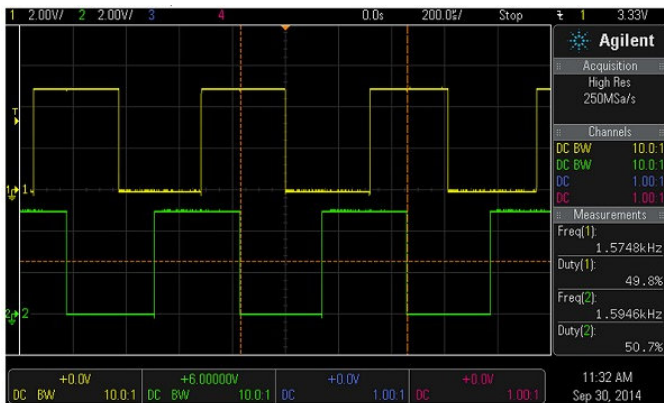


Fig. 4. Square wave captured using an oscilloscope

Having a basic knowledge of how encoders work we now will use the information acquired by the encoders to determine the position of the robot.

The type of position estimation used in our application can be found in scientific literature as *dead reckoning* or *odometry*. The method can vary depending on the information source but the principle remains the same, we usually calculate relative to our previous position.

For mobile robots with differential steering we use the following equations for dead reckoning to determine the X Y position and the heading  $\theta$ :

$$\Delta\theta = 2\pi \frac{R_w}{D} \frac{T_1 - T_2}{T_R} \quad (1)$$

$$\Delta x = R_w \cos(\theta) (T_1 + T_2) \frac{\pi}{T_R} \quad (2)$$

$$\Delta y = R_w \sin(\theta) (T_1 + T_2) \frac{\pi}{T_R} \quad (3)$$

Where:

- $T_1$  represents the number of increments counted on the first encoder
- $T_2$  represents the number of increments counted on the second encoder
- $R_w$  represents radius of the active wheels
- $D$  is the distance between the wheels (measuring from the point of contact of each wheel related to the ground surface)
- $T_R$  is the number of increments counted per revolution

A representation of the parameters we are determining is shown in figure 5.

Fig. 6. LabVIEW Front Panel – Robot position plotting

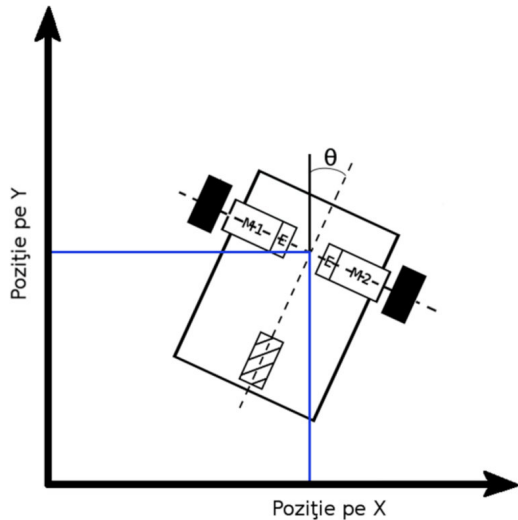


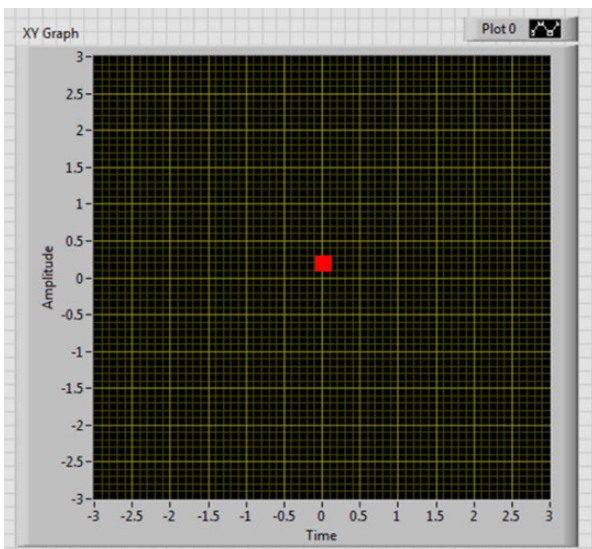
Fig.5. Position parameters of a differential steering robot

After processing the the date acquired from the encoders the position of the robot is plotted on a XY graphic in the LabVIEW Robotics. The XY graphic can be seen in figure 6

The data plotted position of the robot can then aid the operator in positioning the robot without having to have a direct visual contact with the robot.

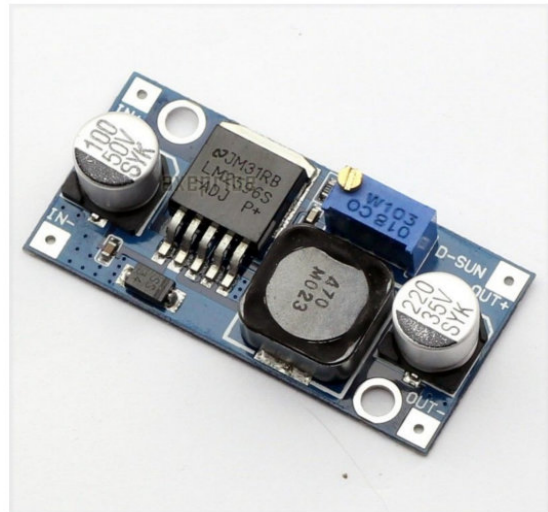
Such system are implemented without needing external beacons and do not need information from external system such as GPS, ultrasonic beacons, laser beacons, etc.

Being a system that is always incrementing its position or in other words calculating its position relative to the previous known position error can accumulate over long periods of time and thus needs error correction after a given time if we want precise positioning of the robots location.



### III. REMOTE CONTROL OF THE ROBOT

The to have a wireless communication we mounted the robot with a wireless TP-Link router and a DC-DC Buck Convert to provide the appropriate voltage (9V) for powering the router. The buck conver can be seen in figure 7 and the



schematic corresponding in figure 8.

Fig. 7 DC – DC adjustable Buck Converter Module

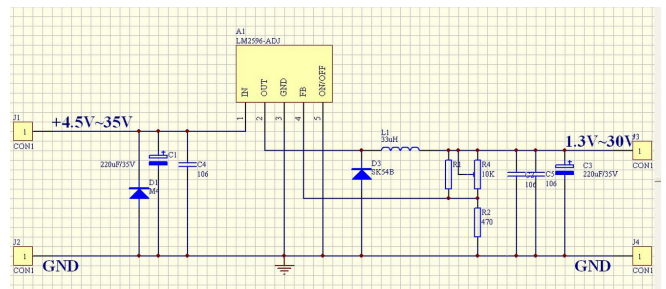


Fig. 8 DC – DC Buck converter Schematic

The buck converter was used to convert the from 12 V provided y the battery to 9V needed to power the wireless router.

To ensure we always have a fixed IP address for the robot the Wireless Router was configured to allocate a specific IP address only to the hardware MAC address of the NI robot, seen in figure 8. This is essential since we do not want to reconfigure and reload the software to the robot every time with a different IP address.

#### IV. VIDEO FEEDBACK

DHCP Clients List			
ID	Client Name	MAC Address	Assigned IP
1	server1	B8-27-EB-CD-8F-16	192.168.1.13
2	NI-sbRIO-9632-175B2DD	00-80-2F-14-22-9D	192.168.1.5

Address Reservation			
ID	MAC Address	Reserved IP Address	Status
1	00-26-5E-3B-E2-51	192.168.1.2	Enabled
2	00-80-2F-14-22-9D	192.168.1.5	Enabled

Fig. 8 – Address reservation for the Hardware Physical Address (MAC)

Having the wireless router mounted on the robot provides another additional 4 Ethernet ports that we can connect standard Network hardware and access it easily. In other words the application does not need a wired connection to the mobile system that would limit the mobility of the robot.

We control the robot using Front Panel of NI LabView. from here the velocity of each motor can be controlled by the user. The control interface can be seen in figure 9.

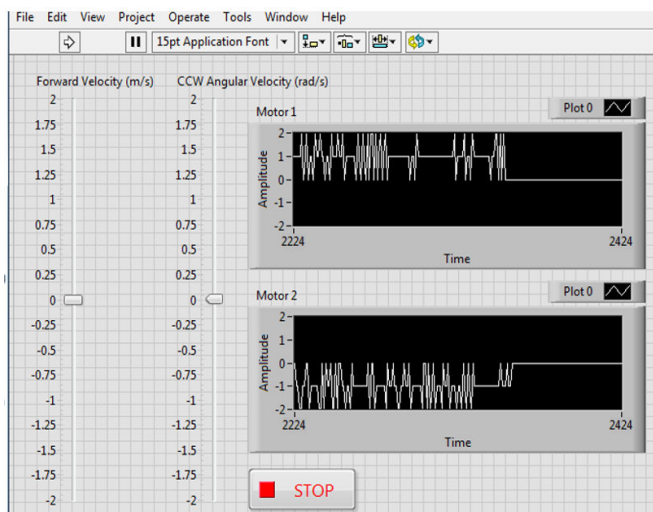


Fig. 9. LabView Control interface – Front Panel

To achieve video the video feedback we have implemented to different solutions. Both of the solutions work via IP and the video streaming can be viewed via Ethernet by any networked device such as laptops, smartphones, tablets and PC's.

Using one Ethernet port from the router we interfaced a standard IP video camera. The solution is easy to implement and the equipment required is relatively cheap.

The second solution for video streaming was implemented using a Raspberry Pi model B rev1 and a USB webcam. The reason for choosing this solution is that it offers more flexibility and a wider application potential. We can use OpenCV for video processing, record video and also host a Web and File server.

The Raspberry Pi was flashed with an ARM based Debian Linux operating system. For doing the flashing of the operating system we used an SD card reader hooked up to a Linux PC and used the following command:

```
user@PC:~> sudo dd if=2016-02-09-raspbian-jessie.img of=/dev/sdb
```

*dd* – is a command-line utility in Unix and Unix-like operating systems. We will use this utility to flash the SD card.

*if* – declares the input file, in our case its the image of our operating system we want to flashing.

*of* – is the output file, in our case its a flash device named *sdb*

After we have set up the operating system on the Raspberry Pi we installed *motion*. This was done using the following command:

```
user@PC:~> sudo apt-get install motion
```

This command will retrieve from the on-line software repository and install the software mentioned.

After this we need to edit the configuration file for motion.

This was done using the *nano* command line text editor with the following command:

```
user@PC:~> sudo nano /etc/motion/motion.conf
```

In this file we set the resolution of the video streaming, the framerate and the port the webcam server can be accessed at.

After setting up the webcam we started the service with the following command:

```
user@PC:~> sudo service motion start
```

After we have started the service we could access the webcam via web browser as shown in figure 10.



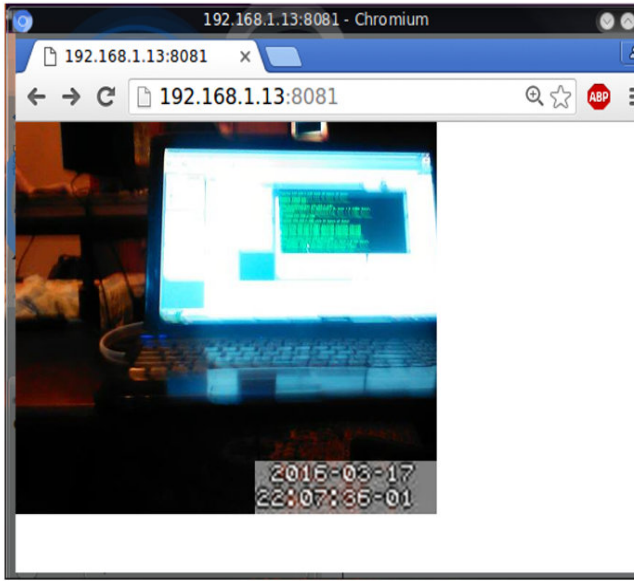


Fig. 10 Video Streaming via Ethernet Webcam as seen from a Web Browser

## V. SYSTEM OVERVIEW

The relation between all components and interconnections are seen in the diagram shown in figure 11. A picture of the physical system assembled with the IP camera is shown in figure 12.

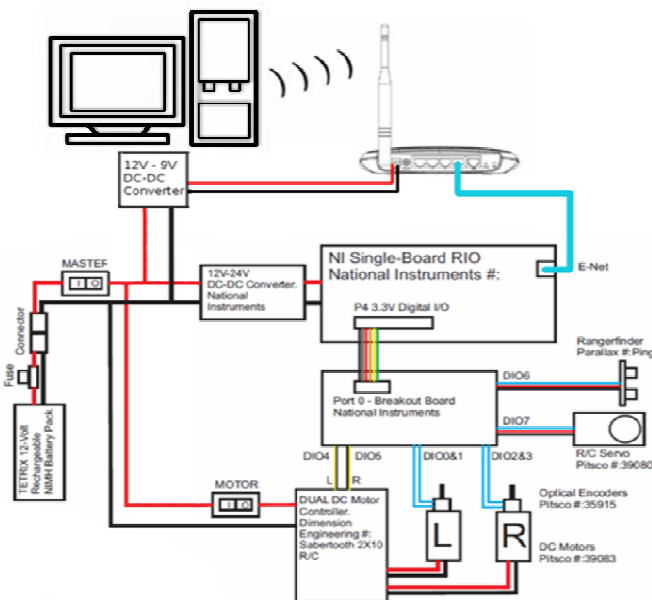


Fig. 11. Block diagram of the robot hardware

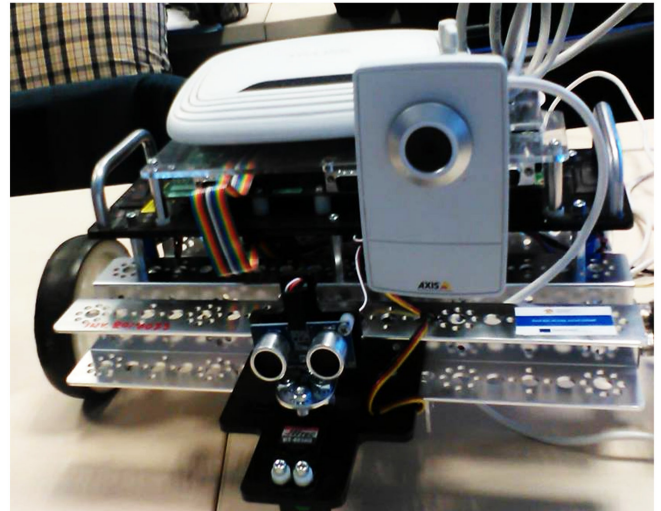


Fig. 12. Assembled Robot Hardware

## VI. AREAS OF USE

The potential use of the application are generally applied to cases where the environment is not suitable for humans. This can be due to small spaces, toxic, radioactive, explosive or other dangerous environment.

Adding appropriate sensors, a few real world applications can be done using the developed system:

- Law enforcement missions such as defusing bombs and explosives
- Explosive Mine detector for finding unexploded land mines
- Mapping and exploring environments that humans could not enter
- Reconnaissance missions
- Investigating and locating gas leaks

## REFERENCES

- [1] Birouaş Ionuț - Flaviu "Dezvoltarea de aplicații pentru LabVIEW™ Robotics Module" bachelor's degree Faculty of I.M.T, Department of Mechatronics, University of Oradea, Oradea, România