Chapter 16

Design and Physical Implementation of Holonomous Mobile Robot – Holbos

16.1. Introduction

Development and miniaturization of electronic components as well as the development of new control algorithms made a great impact on robotics. Robots being built today are becoming even more robust and multifunctional, making them a partner for the future in a number of industries, in medicine, transport and in commercial branches of human activity [BUR 98, DWI 12, FLI 11, MCM 06]. In parallel with the improvement of technologies used in the production and the processes industry, as well as in transport and logistics, the automation and improvement of these processes became an imperative, aiming at larger production rates, but also at better quality products and the reduction in the number of spoiled products. The human error factor is the main reason for these problems occurring in industry today, so the use of robots and robotic systems seems to be an obvious solution. The "3D jobs" (dirty, dangerous and demeaning or dirty, dangerous and difficult) concept, which first emerged as a term that describes the three types of job unwanted by human workers, quickly became a guideline for application-based development of robots and robotic systems [AYR 81, WAL 08, ZAM 11].

Chapter written by Jasmin VELAGIC, Admir KAKNJO, Faruk DAUTOVIC, Muhidin HUJDUR and Nedim OSMIC.

The transport of heavy mechanical and machine constructions and materials in industry is one of the areas in which the application-based development of robots has been quite noticeable in last few decades [FAR 12, FLI 11]. By using pneumatic or hydraulic lifting systems installed on mobile robotic platforms (autonomous or remotely controlled), the lifting and transport tasks became easier and cheaper. A variety of solutions in this area that are already commercially available were thoroughly examined, which led to the main goal of this work – an implementation of a small-scale robotic platform for material or mechanical construction transport with the intention of integration of already existing solutions. Some new solutions will also be introduced [KUR 06].

Most of the existing solutions include remotely controlled robots that incorporate mecanum wheels allowing the omnidirectional movement. The use of these robots still includes the manual control and allows the robot to be used only on flat terrains. The robot being described in this chapter is addressing these issues while incorporating technical ideas and solutions already being used. The guidelines for the development of our robot are as follows:

– The mobile robot will use the omnidirectional movement technology based on mecanum wheels allowing the robot to be used on both flat and tilted terrains.

– The robot will have a retractable platform for transport with leveling option.

– The mobile robot will incorporate manual control based on ZigBee wireless technology, but will also allow the implementation of algorithms for local or centralized autonomous movement of a robot.

– The ultrasonic sensors, encoders and camera vision will provide the input data for algorithms for autonomous movement of a robot.

The locomotion, mechanical and electrical designs, and implementation of the proposed holonomous robot Holbos will be described in the following sections.

16.2. Locomotion of holonomous mobile robot

One of the main features of the holonomous robot is its capability to move in any direction, regardless of its orientation. In other words, the heading and the course of the robot do not need to achieve the same values.

The ability to start moving in any direction represents one of the advantages for this type of design, considering the fact that control for this type of system can be designed in a relatively simple manner [VEL 12]. There are two demands that are set to build the robot with the described capabilities, and both are related to the wheels. The locomotion system that would make this type of robot's movements possible needs to be realized using mechanically complex wheels. Independent speed and direction control for each wheel is another demand in the design of this type of robot. The wheels that are used on this robot are described below.

One type of the mechanically complex wheels that makes the robot holonomous is called mecanum wheel [BLA 90]. Each of these wheels has 12 rubber rollers that are mounted on the edge of the wheel and can rotate freely. The rubber rollers are mounted on the separate shafts that have an angle of 45° relative to the axis of wheel rotation. When the wheel rotates, the point of touch with the ground smoothly moves from one roller to another. Two existing types of these wheels, left-oriented and right-oriented, are always mounted in pair on one shaft of the robot. The way in which the wheels are mounted on a robot is shown in Figure 16.1.

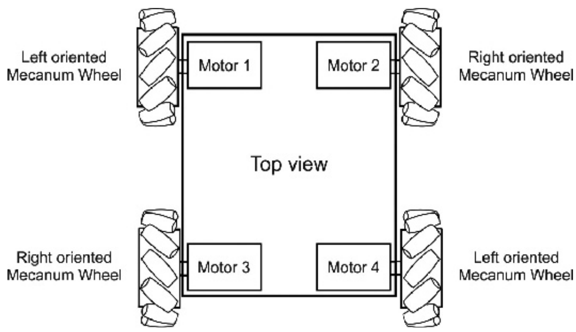


Figure 16.1. *Mecanum wheels and motors on robot*

When all wheels rotate in one direction, clockwise or counterclockwise, the robot is moving forward or backward, respectively. If the wheels on the left side of the robot (according to Figure 16.1) rotate in one direction and the wheels on the right side of the robot rotate in another direction then the robot will rotate in clockwise or counterclockwise direction, depending on the direction of rotation of the wheels. If the wheels in the opposite corners of the robot have the same direction of rotation (wheels one and four rotate in one direction, while wheels two and three rotate in a direction which is opposite to the direction of rotation of wheels one and four), the robot will move to the left or right side, depending on the direction of rotation of wheel pairs. If the wheels in opposite corners of the robot rotate in one direction, while the other two wheels do not rotate, the robot will move diagonally in the direction that is dependant on the direction of rotation of the two wheels. The principle described above is shown in Figure 16.2.

After the dependency of the direction of the robot movement on the mecanum wheel direction of rotation is explained, it is possible to derive a mathematical model of the robot.

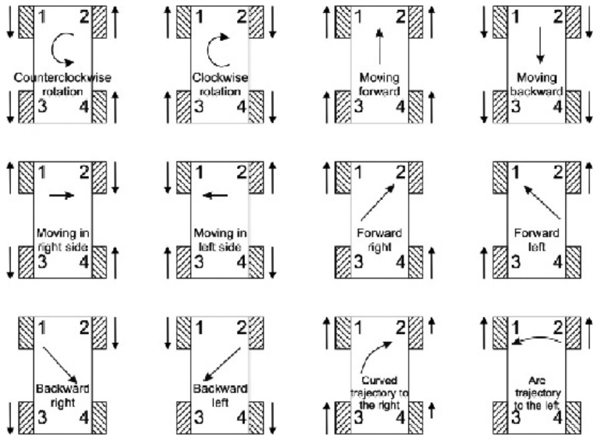


Figure 16.2. *Examples of directions of the movement of the robot*

16.2.1. *Kinematic model of robot*

A kinematic model of holonomous mobile robot in coordinate system xOy (Figure 16.3) can be expressed using the following equation [VEL 12]:

$$x(k+1) = x(k) + v_x \Delta t \qquad [16.1]$$

$$y(k+1) = y(k) + v_y \Delta t \qquad [16.2]$$

$$\varphi(k+1) = \varphi(k) + \omega \Delta t \qquad [16.3]$$

Now, the x and y components of the linear velocity of the robot as well as the angular velocity of the robot need to be determined in every time instant k. The key point in determining those values is the equation for linear velocity of every wheel in the coordinate system that is attached to the robot.

The linear velocity of each wheel in a coordinate system attached to the robot can be expressed as [TLA 08]:

$$v_i^{'} = \mathrm{sgn}(\dot{\theta}_i)K_i r\dot{\theta}_i \qquad [16.4]$$

where K_i is the wheel coefficient that depends on the number of the rollers on the wheel and the distance between two neighbor rollers, r is the diameter of the wheel and $\dot{\theta}_i$ is the angular velocity of ith wheel.

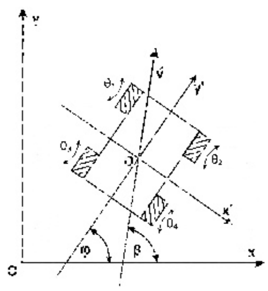


Figure 16.3. *Global and local coordinate system of the robot [TLA 08]*

It is now possible to decompose the linear velocity of the wheels (equation [16.4]) along x' and y' axis of the $x'Oy'$ coordinate system, which yields:

$$v_{ix}^{'} = \mathrm{sgn}(\dot{\theta}_i)K_i r\dot{\theta}_i \cos(\alpha) \qquad [16.5]$$

$$v_{ix}^{'} = \mathrm{sgn}(\dot{\theta}_i)K_i r\dot{\theta}_i \sin(\alpha) \qquad [16.6]$$

where α represents the angle of the rollers relative to the axis of rotation of the wheel.

The total linear velocity components of a mobile robot along x' and y' axis of the $x'Oy'$ coordinate system are sums of velocity components for all wheels. The total linear velocities along x' and y' axis can be expressed as:

$$v_x^{'} = \sum_{i=1}^{4} v_{ix}^{'} = \sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \cos(\alpha) \qquad [16.7]$$

$$v_y^{'} = \sum_{i=1}^{4} v_{iy}^{'} = \sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \sin(\alpha) \qquad [16.8]$$

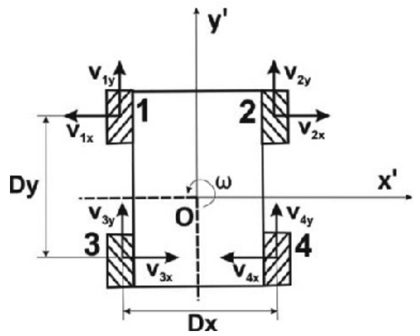


Figure 16.4. *Velocities of the wheels*

It is neccessary to transform equations written in the coordinate system $x'Oy'$ into equations in global coordinate system xOy. In this case, the linear velocity along the x axis in the global coordinate system xOy is described by:

$$\begin{aligned} v_x &= \sin(\varphi)\sum_{i=1}^{4} v_{ix}^{'} + \cos(\varphi)\sum_{i=1}^{4} v_{iy}^{'} = \\ &\sin(\varphi)\sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \cos(\alpha) + \cos(\varphi)\sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \sin(\alpha) = \\ &\sin(\varphi) Kr \cos(\alpha)\left[\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4\right] + \cos(\varphi) Kr \sin(\alpha)\left[\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right] \end{aligned} \qquad [16.9]$$

The equation for the linear velocity along axis y in the global coordinate system xOy can be expressed as:

$$\begin{aligned} v_y &= \cos(\varphi)\sum_{i=1}^{4} v_{ix}^{'} + \sin(\varphi)\sum_{i=1}^{4} v_{iy}^{'} = \\ &\cos(\varphi)\sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \cos(\alpha) + \sin(\varphi)\sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \sin(\alpha) = \\ &\cos(\varphi) Kr \cos(\alpha)\left[\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4\right] + \sin(\varphi) Kr \sin(\alpha)\left[\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right] \end{aligned} \qquad [16.10]$$

The total linear velocity v of the robot can be expressed as a sum of the velocities v_x and v_y from relations [16.9] and [16.10]:

$$\vec{v} = \vec{v}_x + \vec{v}_y \tag{16.11}$$

The direction of the robot can be written as:

$$\beta = arctg(\frac{v_x}{v_y}) \tag{16.12}$$

To determine an angular velocity of a mobile robot, all velocity components that contribute to the angular velocity have to be taken into consideration. For that reason, it is convenient to split all the wheels of the robot into four groups, as shown in Figure 16.4. The robot has non-zero angular velocity if there is any difference in velocity between the wheels on left and on the right side (if the robot is split along the y axis), or if there is any difference in velocity between the wheels on the top and on the bottom side (if the robot is split along the x axis). Consequently, the contribution to angular velocity from every group of wheels is described as follows:

$$\begin{aligned}
&v_{yL} = v_{1y} + v_{3y} \qquad && v_{xD} = v_{3x} - v_{4x} \\
&v_{yR} = v_{2y} + v_{4y} \qquad && v_{xD} = -v_{1x} + v_{2x} \\
&\Delta\theta_y = \frac{v_{yL} - v_{yR}}{Dx} \qquad && \Delta\theta_x = \frac{v_{yD} - v_{yU}}{Dy}
\end{aligned}$$

$$\omega = \Delta\theta = \Delta\theta_x + \Delta\theta_y = \frac{v_{yL} - v_{yR}}{Dx} + \frac{v_{yD} - v_{yU}}{Dy} \tag{16.13}$$

The kinematic model of a holonomous mobile robot can now be expressed as:

$$\begin{vmatrix} x(k+1) \\ y(k+1) \\ \varphi(k+1) \end{vmatrix} = \begin{vmatrix} x(k) + v\cos(\beta)\Delta t \\ y(k) + v\sin(\beta)\Delta t \\ \varphi(k) + \omega\Delta t \end{vmatrix} \tag{16.14}$$

Equation [16.14] can be rewritten in this form:

$$\begin{aligned}
&x(k+1) = x(k) + \\
&\left\{s\text{in}(\varphi)Kr\cos(\alpha)\left[\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4\right] + \cos(\varphi)Kr\sin(\alpha)\left[\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right]\right\}\Delta t
\end{aligned} \tag{16.15}$$

$$y(k+1) = y(k) + \left\{\cos(\varphi)Kr\cos(\alpha)\left[\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4\right] + \sin(\varphi)Kr\sin(\alpha)\left[\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right]\right\}\Delta t \quad [16.16]$$

$$\varphi(k+1) = \varphi(k) + \left(\frac{\mathrm{sgn}(\dot{\theta}_1)\dot{\theta}_1 + \mathrm{sgn}(\dot{\theta}_3)\dot{\theta}_3 - \mathrm{sgn}(\dot{\theta}_2)\dot{\theta}_2 - \mathrm{sgn}(\dot{\theta}_4)\dot{\theta}_4}{Dx} + \frac{\mathrm{sgn}(\dot{\theta}_1)\dot{\theta}_1 + \mathrm{sgn}(\dot{\theta}_3)\dot{\theta}_3 - \mathrm{sgn}(\dot{\theta}_2)\dot{\theta}_2 - \mathrm{sgn}(\dot{\theta}_4)\dot{\theta}_4}{Dy}\right)\Delta t \quad [16.17]$$

Equations [16.15]–[16.17] represent the kinematic model of an holonomous mobile robot. In the following section, a mechanical design of the proposed holonomous mobile robot will be described.

16.3. Mechanical design

Since the robot building is done on a small scale and the accent was not on the mechanical design, the decision was made to use a commercially available plastic box with a reinforced bottom. This made the robot durable, but at the same time lighter. It also allowed easier mounting of other mechanical and electronic parts. In the four corners of the lower part of the mobile construction four motors were mounted. The mecanum wheels were directly mounted onto the motors, together with encoder sensors allowing the measurement of the speed of each of the wheels. Inside the mobile robot, a 12 V battery together with the printed circuit boards was mounted. Four linear actuators were also mounted in the four corners of the lower part of robot. The upper part of the robot, which can be easily disconnected in order to access the interior of the robot, is used as a platform. Although the main purpose of this platform was to implement and test the platform control algorithms and not to lift heavy objects, it turned out that it allowed a significant load to be lifted and carried (around 5 kg). On the bottom side of the platform, at its center, an inertial measurement unit (IMU) was mounted, allowing the sensing of the tilt angle of the platform. Four light-emitting diodes were also added at corners of the platform. These light–emitting diodes blink when the platform is being manually controlled. The ultrasonic sensors were mounted on a steel support mounted on the robot. These sensors were installed in 4-2-2-4 formation (four on the front and back and two on the left and right sides), sensing the robots distance from the obstacle. The steel support for the kinect camera was also mounted on the back side. This makes the development, testing and use of complex, vision-based, robot control algorithms possible on this robot. The three-dimensional model of the robot and its picture are shown in Figure 16.5.

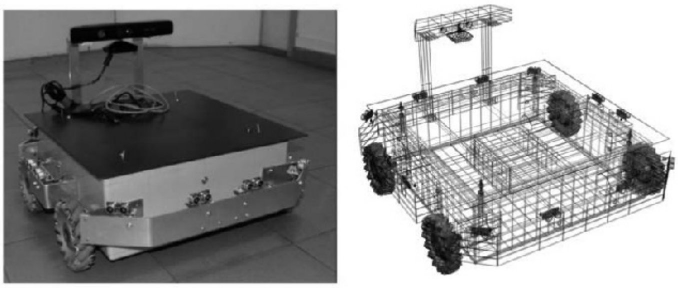


Figure 16.5. *Picture and 3D model of Holbos robot*

For the wireless manual control of the robot, the Sony PlayStation joystick was used. Together with the ZigBee wireless communication module and a display, it was connected with the Arduino Uno development board. The controller is powered by 9 V battery and is shown in Figure 16.6.

Figure 16.6. *Manual control module*

16.4. Electrical design

16.4.1. *Hardware components and software implementation*

The hardware structure of the robot is designed as a distributed control system. There are three separate modules that are responsible for controlling different parts of the robot (different subsystems). These modules are interconnected using a RS-232 serial or the ZigBee wireless communication, depending on the chosen control module type. The block structure of the hardware system of the robot is shown in Figure 16.7. This section, which deals with the description of the modules, is divided into four parts as follows:

– main module (sensors and communication);

– manual control module (joystick);

– platform control module;

– motor control module.

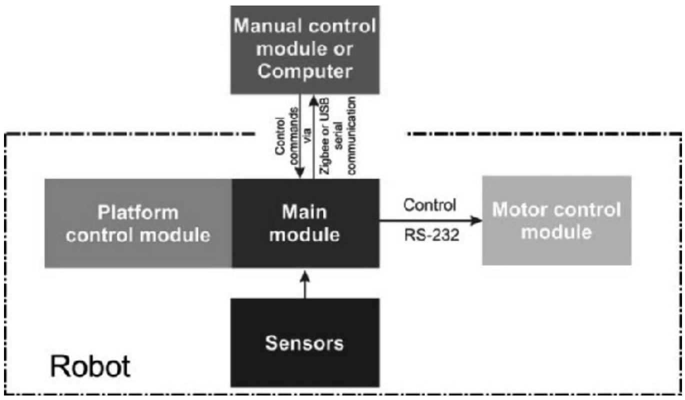


Figure 16.7. *Block structure of realized system*

This type of design allows for flexibility in terms of controlling the robot. There are two possible modes of robot operation: manual remote control and automatic control. To control the robot manually, a special joystick with ZigBee wireless communication was made. ZigBee wireless communication makes remote controlling of the robot possible. By placing the laptop computer, or some other piece of hardware with some control algorithms, on the robot platform, and connecting it to the main module, the robot can become autonomous.

16.4.1.1. *Main module*

The main module got its name because it represents the hardware device that interconnects all other modules on the robot. All sensors that are mounted on the robot are also connected to this board. During the planning stage of the project, it was found that this device has to have at least four hardware modules for serial communication and a large number of digital and analog inputs. After reviewing several possible options, the decision was made to use an Arduino Mega board for this purpose. Not only does the board fulfill all the demands, but it also represents a low-cost solution for this type of use. With four serial communication modules and a total of 60 analog and digital inputs, the Arduino Mega board is an ideal solution for the main module of the robot.

The main module is responsible for several important tasks. The first task is to receive commands from the joystick or computer and to send them to the motor control module or to use them as control commands for the platform control module. These commands can be generated by a human operator using the joystick, or by a computer, and they make the manual control of the robot and platform movements possible. In the case of the computer control of the robot, the main module has to collect data from various sensors and send them back to the computer. On the basis of these data, the computer can run different algorithms and, based on that, different movement commands can be issued and sent to the main module. There are four types of sensors that are directly connected to main module:

– an ultrasonic range finder;

– an incremental encoder;

– a 2-axis gyroscope;

– a 3-axis accelerometer.

16.4.1.1.1. Ultrasonic range finder

There are 12 ultrasonic range finder sensors mounted on the robot body. Two sensors are mounted on each side, while four sensors are placed in the corners of the robot at an angle of 45° relative to the line perpendicular to any side of robot. The way that sensors are mounted on the edge of the robot is shown in Figure 16.8. Sensors are labeled with numbers – from one to 12. The numbers show the order in which the data from the sensors are collected (sensor number one is checked first, then sensor number two, etc.). Sensors can measure the distance from the objects in its environment in the range of 40–400 cm from the robot. The main module collects the distance data from every sensor and, if needed, sends them to the computer.

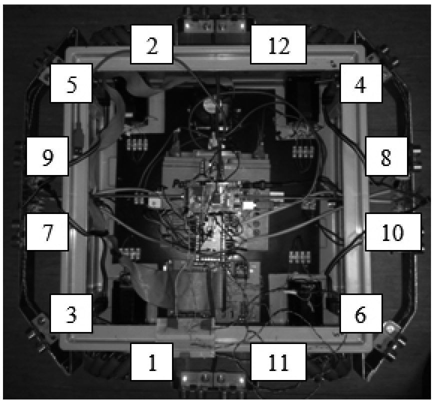


Figure 16.8. *Ultrasonic range finders on robot*

16.4.1.1.2. Incremental encoders

Each wheel on the robot is equipped with the incremental encoder. Signals from encoders are fed directly to digital inputs of the main module, where the frequency of every signal is calculated and, if needed, sent to the computer. The frequency of the encoder signal is directly proportional to the speed of the rotation of wheels. The data from encoders is very valuable in most applications where localization of the robot is needed.

16.4.1.1.3. Kinect camera

Kinect is a device that was developed by Microsoft for use in their game console Xbox 360. It has a Charge-Coupled Device (CCD) Red Green Blue (RGB) camera, 3D depth sensor and a microphone array. The possibility of acquiring both 2D and 3D images of the environment in the same time made this camera very convenient for use in robotic applications. The camera connects to a computer using a USB serial port, and if appropriate drivers are used, it is possible to get raw data from the camera.

16.4.1.1.4. Communication

Every message, that needs to be sent from one module to another, first has to be sent to the main module of the robot, which then forwards the message to its destination. Some hardware components that are used for communication were already mentioned in the previous sections. In this section, a detailed description of the messages format and the communication protocol will be provided.

The establishment of communication between the main module and the joystick or computer is the first task to be performed after turning on the robot. When the robot is switched on, the main module expects to receive an initialization message from the joystick. The initialization message consists of six bytes, of which the first byte is a hash sign "#"". After this sign, there are four zeros and letter "E". When the message with this content is received by the main module, it replies with the same format message. These two messages differ only by the last character. The main module replies with the message that ends with character "O", which means that communication is established or that communication is "OK". When this message is received by the joystick, notification about successfully established communication appears on the LCD display, so the operator can see the result of the initialization stage. It is important to point out that the format of the message that the joystick sends to the main module is always same and has the following form: "#n1n2n3n4C". Every message consists of the hash sign, four numbers (n1, n2, n3 and n4) and a character. In this section, the meaning of the various characters and messages that are sent will be explained.

Before the communication is further explained, it is important to say few things about the modes of operation of the robot. In the previous section, it was stated that the robot has two modes of operation, dependent on the device that controls the robot (joystick or computer). These modes of operation were called manual and automatic and they refer to the presence of the human operator. These two modes should not be mistaken with the modes of operation of the platform, which are also called automatic and manual. In the automatic mode of operation, the platform is controlled by the platform control module, whereas in the manual mode the platform is controlled by the human operator or computer. A detailed description of the way that the platform works will be provided in one of the following sections.

After initialization, the operator can choose between two modes of operation of the robot – control of the robot movements or platform control. The selection between these two modes of operation can be done by pressing the buttons "Start" or "Select" on the joystick. If the operator presses the "Start" button, the message that ends with the character "S" is sent to the main module, which means that the robot movement control mode is chosen. If the operator presses the "Select" button, the message that ends with the character "s" is sent to the main module, which means that the platform control mode is chosen. The mode selection message can be sent out at any time whilst operating the robot, when a change in the mode of operation is needed. The described algorithm can be written as pseudocode as shown in Table 16.2.

In the robot movement control mode, the message from the joystick carries four set-point values for the speeds of every wheel. The message ends with the character “A”. The speed set-points can be in a range [0, 255], where the range [0, 127] represents a positive direction and the range [128, 255] represents a negative direction of rotation of every wheel. This message can be graphically represented as shown in Figure 16.9. Four numbers that are sent in every message are ignored in every other mode of the robot operation, and the message is decoded using the last character. Different characters mean a different action.

Byte	1	2	3	4	5	6
Character	#	Speed1	Speed2	Speed3	Speed4	A

Figure 16.9. *Movement control mode message*

In the platform control mode of operation, there are four different messages that can be sent to the robot. The messages differ only by the last characters. Four numbers in every message are equal to zero. The messages for manual platform control are shown in Table 16.1, together with the explanation of every message.

Message	Meaning
#0000G	Raise the platform
#0000P	Lower the platform
#0000R	Raise one side of the platform
#0000B	Raise other side of the platform

Table 16.1. *Messages for manual platform control*

In the case of computer control of the robot, a communication protocol can be easily expanded in the way that the main module sends the sensor data to the computer. When the computer is placed on the platform, it connects to the main module using a USB cable, which allows full duplex communication between computer and robot. With full access to sensor data, various advanced control algorithms can be realized and the robot can be turned into fully autonomous vehicle.

Startup – program
while "Initialization" message is not received
Determine if message is "Initialization" message
end while
Send message: "Communication established"
while "Mode selection" message is not received
Read message
Determine if message is "Mode selection" message
end while
Determine the mode of operation
if mode of operation is "Control of robot movements"
Control_of_robot_movements()
end if
if mode of operation is "Platform control"
Platform_control()
end if
End

Table 16.2. *Algorithm that runs when the system is powered*

16.4.1.2. *Manual control module (joystick)*

As stated in the previous section, a robot has two modes of operation, manual and automatic. In automatic mode, the robot is in control of a computer and can perform tasks autonomously. However, automatic operation is an extension to the original design of the robot, which is primarily designed to be operated by human operator. Thus, the robot is equipped with a device that makes remote operation by a human possible.

Some of the goals that were set for the joystick design, in addition to the possibility of remote operation, were the intuitive control and ergonomic design of the joystick. For these reasons, it was decided to use the type of joystick that is typical for game consoles, such as PlayStation. The joystick is equipped with the Arduino Uno board and ZigBee module for wireless communication. The Arduino board receives messages from the joystick and transforms them into instructions that are sent to the main module of the robot. The modes of operation of the robot and some other important messages are shown on an LCD display that was also mounted on the joystick and connected to the Arduino Uno board. The block structure of the manual control module is shown in Figure 16.10. The algorithm, described in words, is shown as pseudocode in Table 16.3.

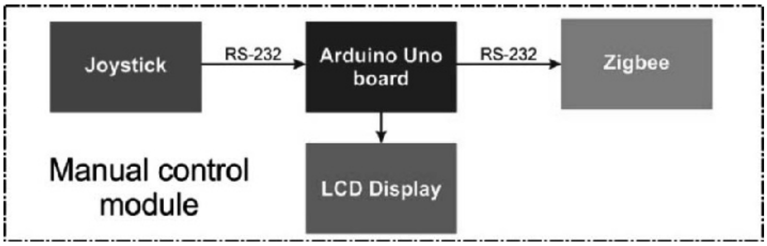


Figure 16.10. *Structure of manual control module*

Joystick_Algorithm
while "Acknowledge message" is not received
Send the "Initialization message"
end while
Display "Communication established" text on LCD display
while 1
if operator pressed the button
if "Change the mode of operation" message sent
Change text on display
end if
Send the message to main module
end if
end while
End

Table 16.3. *Algorithm that runs in manual control module*

16.4.1.3. *Motor control module*

Mechanical realization of the locomotion system was explained in detail in previous two chapters. This section deals with the hardware realization of a control module for motors. As stated in previous chapters, the robot has four wheels and four motors in total, with every wheel mounted directly on the shaft of the motor. This approach provides the ability to control every motor separately and thus gives more flexibility when defining the robot's trajectory. The block structure of the module that is responsible for motor control is shown in Figure 16.11.

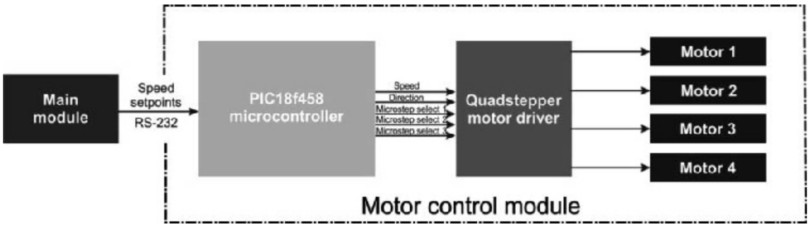


Figure 16.11. *Motor control module*

The module consists of two main components, the microcontroller-based board and the motor driver.

16.4.1.3.1. Microcontroller-based board

The main task of the microcontroller is to receive speed set-points for every motor, and to generate control signals for the driver board. Serial RS232 communication is used to transmit speed set-point values from the main module to a PIC18f458 microcontroller. Five signals need to be generated for each motor, a total of 20 signals for four motors, in order for the motor driver to work properly. Two of the five signals are speed and direction signals for each motor. The speed signal is in the form of squares with an amplitude of 5 V. The frequency of the squares is directly related to the speed of the motor. Every rising edge of the signal causes the motor to rotate by one step (1.8°) in a clockwise or an counterclockwise direction. The speed of each motor is directly proportional to the frequency of the control signal; the higher the frequency of the control signal, the higher is the speed of rotation of the motor. The direction signal dictates the direction of rotation. If the signal is in a low state, the direction is clockwise, and vice versa – if the signal is in a high state, the direction is counterclockwise.

16.4.1.3.2. Motor driver

For more simple control of the motors, a Quadstepper motor driver, and a single board with four motor drivers was used. As stated above, every driver requires five signals for every motor. The purpose of speed and direction signals is explained in the previous section. The other three signals (microstep select) are used to determine the mode of operation for the driver.

There are five different modes of driver operation (full step, half step, quarter step, eighth step and sixteenth step). Quarter, half and full step microstepping modes are used. When starting the rotation of a stepper motor, it is convenient to start with lower speeds. Quarter and half microstepping modes are used in this respect. The speed of the stepper motor can be lowered by choosing one of the microstepping modes of operation, while the frequency of the speed signal can remain constant and equal to the set-point value of the speed. When the stepper motor is starting, first the quarter step mode is on. After a quarter of a second, the mode changes to half step and after another quarter of a second the mode of operation becomes the full step mode. The change of the modes of operations ensures the rotation of the motor, even when the robot has to carry an increased amount of load.

The way in which the main module sends the set-points to the motor control module can be represented as shown in Table 16.4.

Control_of_robot_movements

```
while "Change the mode of operation" message is not received
 if message is received
     Read the message
         if message contains commands for motors
             Send control message to Motor control module
       end if
 end if
 end while
Control_of_robot_platform()
End
```

Table 16.4. *Algorithm for controlling robot's movements*

16.4.1.4. *Platform control module*

The position of linear actuators and the IMU unit is described in section 16.3. The Firgelli L12-50-100-12-I multipurpose linear actuators were used. These actuators include the on-board controller that allows servo (0–5 V), pulse-width modulation (PWM) (Transistor-transistor logic (TTL) level signal) or current position (4–20 mA) control and have a stroke of 0–5 cm. The actuators output voltage signal (0–3.3 V) can be used to implement a closed-loop position control. The on-board controller is capable of discovering the used input control signal without any manual adjustments.

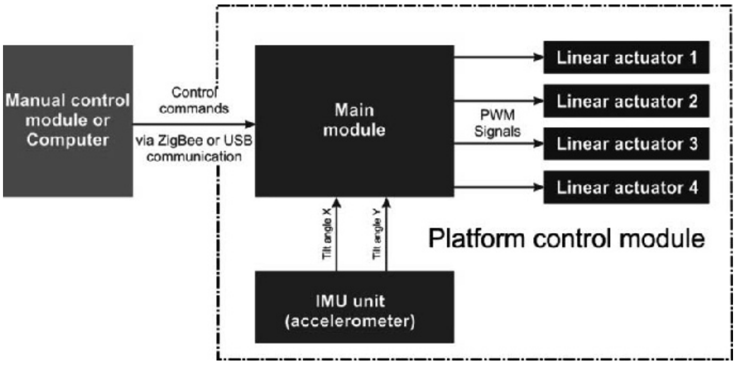


Figure 16.12. *Platform control module*

The IMU unit used for platform leveling is a five degree of freedom (DoF) unit that incorporates the IDG500 gyroscope and the ADXL335 accelerometer. This IMU unit, besides its intended use, can be used for localization algorithms when the platform is stationery by measuring roll, pitch and the tilt angle, as well as the X, Y, Z coordinates of the robot.

The platform control algorithm generates the PWM signals for each of the four linear actuators. These signals are TTL level square signals and have the frequency of 500 Hz. The stroke of 0–5 cm is corresponding to the 0–100% of PWMs duty cycle. Since the task was to keep the platform level, regardless of the tilt angle of the terrain the robot was moving on, the IMU unit was used to measure the tilt angle of the platform. Because of the nature and logic of this task, the exact data on tilt angle were not needed and since the linear accelerations of the robot in practice were negligible, the data given by the accelerometer were sufficient for the control implementation.

The block scheme of the platform control module is given in Figure 16.12. As can be seen in the figure, the control of platform movement is received via ZigBee wireless communication or the USB port of Arduino Mega unit. The command can be received from a remote controller or from any other unit with an on-board control algorithm that is connected to the main module. Based on commands received by the main module, which were already explained in the previous section, there are two modes of operation – robot movement control mode (selected by pressing the “start” button on the joystick) and the platform control algorithm mode (selected by pressing the “select” button on the joystick). In platform control algorithm mode, the movement of platform depends on the message that is sent from the user interface (joystick or computer) to the platform. This mode is used only for manual control of the platform and can be written in pseudocode as shown in Table 16.5. When the robot is in robot movement mode, the platform leveling is autoregulated based on the readings from the accelerometer. The implemented algorithm is described by pseudocode as shown in Table 16.6.

Platform_Manual_Control

```
while "Change the mode of operation" message is not received
 if message is received
    Read the message
       if message contains commands for actuators
          Send control message to Platform control module
     end if
 end if
end while
Control_of_robot_movements()
End
```

Table 16.5. *Algorithm that runs when platform control module is in manual mode*

In Table 16.6, the following abbreviations are used:

– DIV_X is digital value of the x component of acceleration.

– DIV_X0 is digital value of the x component of acceleration at 0° angle.

– DIV_Y is digital value of the y component of acceleration.

– DIV_Y0 is digital value of the y component of acceleration at 0° angle.

– SOA is sensitivity of accelerometers.

The power supply voltage of the accelerometer used is 5 V and its sensitivity is 0.33 V/g, where V is voltage and g is the relative gravitational acceleration. Constant 1,023 in denominators is the number of quantized levels of a 10 bit analog-digital converter. The equations for angle calculation are derived from a trigonometric description of the gravitational acceleration vector and its components, as can be seen in Figure 16.13.

Platform_Automatic_Control	
1	Get the digital value of the analog signals of current X and Y components of gravitational acceleration
2	Calculate the value of X and Y components of gravitational acceleration by using equations: $g_x = \frac{DIV\_X - DIV\_X0}{1023 \times SOA}$ $g_y = \frac{DIV\_Y - DIV\_Y0}{1023 \times SOA}$
3	Calculate the value of tilt angles (in degrees) by using equation: $angle\_X = \arcsin(g_x)\frac{180}{\pi}$ $angle\_Y = \arcsin(g_y)\frac{180}{\pi}$
4	***if*** any of the angle values is greater than $\pm 1^{\circ}$
5	Choose which linear actuators are to be extracted based on angle values and signs
6	***end if***
7	Extract the chosen linear actuator by 1 mm

Table 16.6. *Algorithm for automatic platform control*

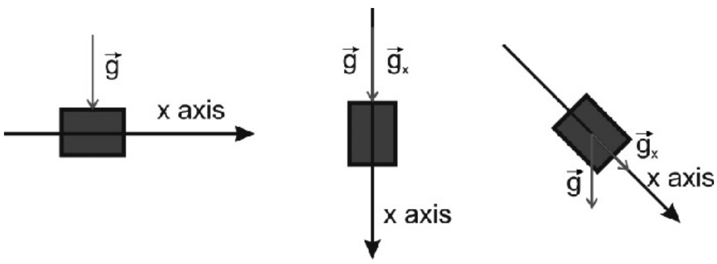


Figure 16.13. *Components of gravitational acceleration vector*

When the referent axis for the measurement of the angle (X axis in Figure 16.13) is parallel to the Earth's surface, the value of relative gravitational acceleration is 0. When the referent axis is perpendicular to the Earth's surface, relative gravitational acceleration is 1. In any other position, the component of gravitational acceleration on each of the referent axis will have a value between

0 and 1. These values are then measured by the accelerometer, and based on them, tilt angles can be calculated as described in step 3 of the pseudocode shown in Table 16.6. By using simple trigonometric operations, it is possible to determine the sign of the tilt angle of the platform relative to the ground. The value and the sign of the calculated angle is then used in the decision-making process which results in instructions that consist of extraction or retraction of each of the linear actuators. The ±1 degree dead zone was chosen so the algorithm would not be influenced by small vibrations and smaller tilt angles. The algorithm described is repeated periodically (5 ms period).

16.5. Results

After assembling the robot, a thorough testing of the performance of the overall system was done. Tests were mostly conducted by an human operator, with the robot in the manual mode of operation. During the tests, several important parameters that describe the robot's behavior were observed.

The first parameter observed was the degree of autonomy of the mobile robot, which was determined by measuring the time that it takes the battery to discharge to the level at which the normal operation of the robot is impossible. The robot's battery discharge time lasted from 5 to 7 h of operation, dependant on the load that was on the robot platform. Increased battery drain and current as well as the decrease of the robot's operational time occur when the robot is used in an environment with angular slopes. When the robot is moving up the slope, besides the increased motor current, there is also a significant current being drawn by the linear actuators that are keeping the load of the robot in a horizontal position. These are some identified factors that affect the battery life.

A second parameter was the maximum distance between the operator and the robot on which the uninterrupted control of the robot is still possible. The control loss occurs due to the limited range of the wireless ZigBee transceivers on the joystick and on the robot. Maximum range at which the robot was still operable was about 30 m. This experiment was conducted in an indoor environment with reinforced concrete walls that significantly weakened the signal. If the robot is used in an outdoor environment, the range is increased up to 100 meters. It is possible to achieve these ranges because wireless transceivers with external antennas are used.

The main purpose of the robot is to transport and lift heavy materials and objects, and for that reason it was very important to check the maximum load that the robot can carry. Although only a prototype built on a small scale, a significant amount of payload can be carried around on the robot's platform, without affecting the performance of the robot. The upper limit for the payload is set at 5 kg, although the robot can carry even more, but with noticeable effects on its performance.

After the experiments in which the behavior of the robot was monitored and observed, the robots capability of executing some simple tasks autonomously was put to the test. To show an autonomous operation of the robot, a simple experiment that includes operation of almost all systems of the robot was designed. The scenario of the experiment is the following:

The robot is placed in a controlled environment (3 m × 3 m polygon), together with the red colored, round-shaped target, glued to one of the walls of the polygon. The task was to have the robot rotate on the spot, until the red target is detected. After detection, the robot should move toward the target and stop at the distance of approximately 20 cm from the wall. In the end, the robot should calculate the distance from its initial position to the place where it stopped moving. To do such an experiment, a laptop computer was placed on the platform and connected to a kinect camera and to the main module of the robot. The image from the kinect camera is used to detect and identify the marker (target), while the ultrasonic range finders were used to determine the distance from the wall. A simple color detection algorithm was implemented on the computer, using Matlab® program. Besides color detection, a simple control algorithm, which sends the commands to the main module, was also implemented in Matlab®. In this case, the main module was providing the data from the encoders and the ultrasonic range finders to the computer. The computer used these data to determine the distance of the robot from the wall as well as the total covered distance. Some images that were captured during the experiment are shown in Figure 16.14.

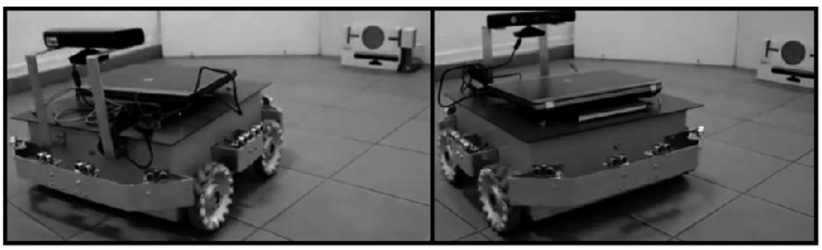
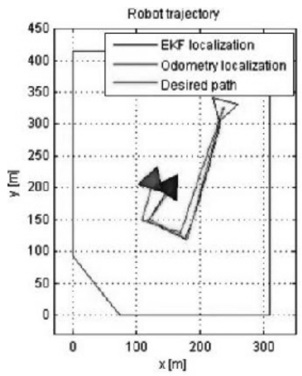
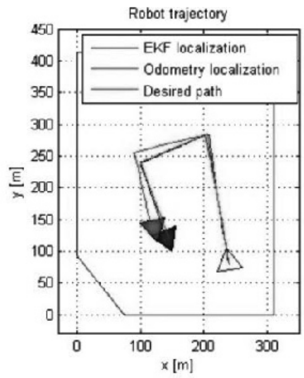


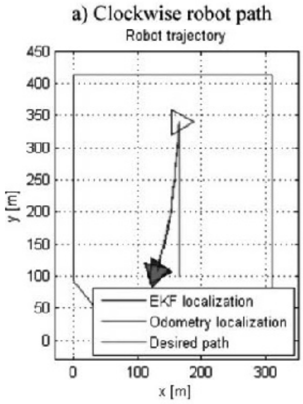
Figure 16.14. *Robot performing tasks autonomously*



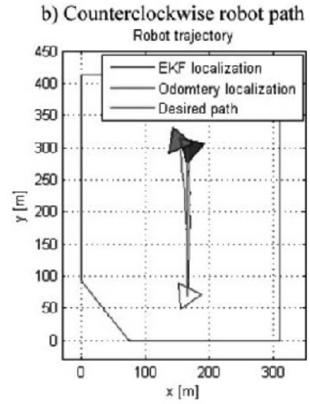
a) Clockwise robot path



b) Counterclockwise robot path



c) Robot moving to the right



d) Robot moving to the left

Figure 16.15. *Experiments with localization of the robot*

With all sensors and systems checked and proved working, it was possible to implement some more complex algorithms. One of the main problems in mobile robotics is localization or determining the position and orientation of mobile robots [HAB 07]. Thus, the localization algorithm was implemented in two ways (odometry- and landmark-based localization using an extended Kalman filter). In total, four experiments with localization were conducted. In every experiment, both the odometry- and landmark-based localization algorithms were running simultaneously. Figure 16.15 shows the results of the experiments after running tests with both algorithms. The first two experiments (Figures 16.15(a) and (b)) refer to the robot moving in the way in which the non-holonomous robots move, while the other two experiments (Figures 16.15(c) and (d)) emphasize the robot's possibility for holonomous movements. Transparent triangles refer to the start point of the robot, while dark gray and light gray triangles refer to the end point of the robot. The dark gray triangle (and dark gray trajectory) is the result from landmark-based localization, while the light gray triangle (and light gray trajectory) is the result from odometry localization. As expected, the results gained from the experiments with localization differ, and also, the position and orientation of the robot obtained from landmark-based localization with extended Kalman filter (EKF) are more accurate than the results obtained using only the odometry localization. The reasons for such results lie in the difference of the operation of the two algorithms and also in the number of sensors required by the algorithms. A detailed analysis of the localization results is beyond the scope of this chapter.

16.6. Conclusion

The design and building of a mobile robot platform – HOLBOS explained in previous sections – offers a multipurpose lifting and transportation solution incorporating most of the commercially available solutions in this area. Although the robot is built on a small scale, it proves its effectiveness throughout the conducted tests and experiments.

A user-friendly manual control interface makes the robot easy to operate for an average user. Its robustness and the significant weight that the robot can lift and transport, makes it operable for industrial conditions, regardless of its small-scale design. The mecanum wheels allow the omnidirectional movement of the robot that significantly improves its handling in tight areas that are often encountered in industry or on assembly lines. The autonomy given by a long-life battery power supply and wireless control allows the user to manage the transport tasks with ease, intuitively, and without any special training.

When compared to commercially available solutions, few improvements are noticeable:

– Although cheap to build, the Holbos robot represents a complete and functional solution capable of performing multiple different tasks simultaneously, as described in this chapter.

– The autonomous leveling of the platform, while simultaneously carrying the load, expands the borders for possible areas of the robot's use. This makes it capable of working not only on flat, but also on sloped surfaces.

– The use of a kinect camera and ultrasonic sensors, together with the possibility of a computer-based control, gives the possibility of an autonomous, vision-orientated control of the robot. This also makes a good base for the implementation and testing of other control algorithms for the autonomous control of the robot, based on data from the kinect camera or the ultrasonic range finder sensors.

– The wireless, ZigBee-based, control can be used to reprogram the robot's tasks and its paths from a central control computer or system, therefore allowing groups of the same or similar robots to be programmed and controlled at the same time, enhancing their performance and effectiveness. This not only saves money being spent on labor power, but also saves time needed for lifting and the transportation tasks in production. It also makes these tasks less dangerous and easier for humans.

– Holbos is intended to be a platform for further development, not only for the development of algorithms, but also for different tasks that the robot should fulfill, either in industry or in some other areas.

– Based on these improvements, the Holbos robot is already being used as a platform for development and testing of different control algorithms in the Faculty of Electrical Engineering in Sarajevo, therefore becoming the didactic and research and development tool.

16.7. Bibliography

[AYR 81] AYRES R., MILLE S., *The Impacts of Industrial Robots*, Carnegie-Mellon University, Pittsburgh, PA, 1981.

[BLA 90] BLACKWELL M., The URANUS mobile robot, Technical Report CMU-R-TR-91-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 1990.

[BUR 98] BURGARD W., CREAMERS A.B., FOX D., *et al.*, "The interactive museum tour – guide robot", *Proceeding of the 15th National Conference on Artificial Intelligence*, Madison, WI, 1998.

[DWI 12] DWIVEDI J., MAHGOUB I., "Robotic surgery – a review on recent advances in surgical robotic systems", *Florida Conference on Recent Advances in Robotics*, Boca Raton, FL, May 2012.

[FAR 12] FARES K.N., EL-HAGRY M.T., EL-KOUSY A.A., "Mobile robot navigation in warehouses for materials handling", *4th Saudi Technical Conference and Exhibition*, STCEX 2006, vol. 3, pp. 144–150, 2006.

[FLI 11] FLIPSE M., Altering and improving Kiva, some suggestions for improvement of the current Kiva system, Technical Report, Vrije Universiteit, Artificial Intelligence Department, 2011

[HAB 07] HABIT M.K., "Real time mapping and dynamic navigation for mobile robots", *International Journal of Advanced Robotic Systems*, vol. 4, no. 3, pp. 323–338, 2007.

[KUR 06] KUROI K., ISHIHARA H., "A four-leg locomotion robot for heavy load transportation", *International Symposium on Automation and Robotics in Construction* (ISARC), Beijing, 9–15 October 2006.

[MCM 06] MCMAHAN W., CHITRAKARAN V., CSENCSITS M., *et al.*, "Field trials and testing of the OctArm continuum manipulator", *Proceedings of the 2006 IEEE International Conference on Robotics and Automation,* Orlando, FL, May 2006.

[TLA 08] TLALE N., DE VILLERS M., "Kinematics and dynamics modelling of a mecanum wheeled mobile platform", *Council for Scientific and Industrial Research*, Pretoria, South Africa, 2008.

[VEL 12] VELAGIC J., *Mobilna Robotika*, Sarajevo, 2012.

[WAL 08] WALLEN J., The history of the industrial robot, Technical Report from Automatic Control at Linkopings University, Linkoping, Sweden, May 2008.

[ZAM 11] ZAMBACH S., LOMONACO L., The influence of robots on the human society, Technical Report ID-7219, Roskilde Universitet, Denmark, 2011.