

SOFTWARE FOR CONTROL OF ROBOTIC MANIPULATOR

INTRODUCTION

Robotics is a very common and discussed problem recently. Robotic arms and manipulators are used in industry, transport or medicine. They often substitute people where the operations are dangerous, repeating. Robotics can be divided into three parts:

- theoretical robotics search principles and possibilities
- experimental robotics proves principles and builds models
- applied robotics builds robots for industry, medicine etc.

They are used for operations, such as welding in automotive industry, or for dosage in chemical industry or medicine. Robots can also be equipped with cameras and wheels. These can be used in sites of disasters as devices that save lives.

1 DEVICE DESCRIPTION

This device is using Lego Mindstorm and MATLAB. Lego NXT Brick is using 32-bit microcontroller and can be connected up to three motors and four sensors as touch, light, colour or ultrasonic sensor [Hlinovský et al., 2010]. The NXT brick can be connected to PC via USB 2.0 or Bluetooth.



Fig. 1 – NXT Brick with accessories

Lego develops graphical interface, where user can easily program the brick using block schemes. Otherwise MATLAB and free toolbox from RWTH Aachen University can be used to program the brick.

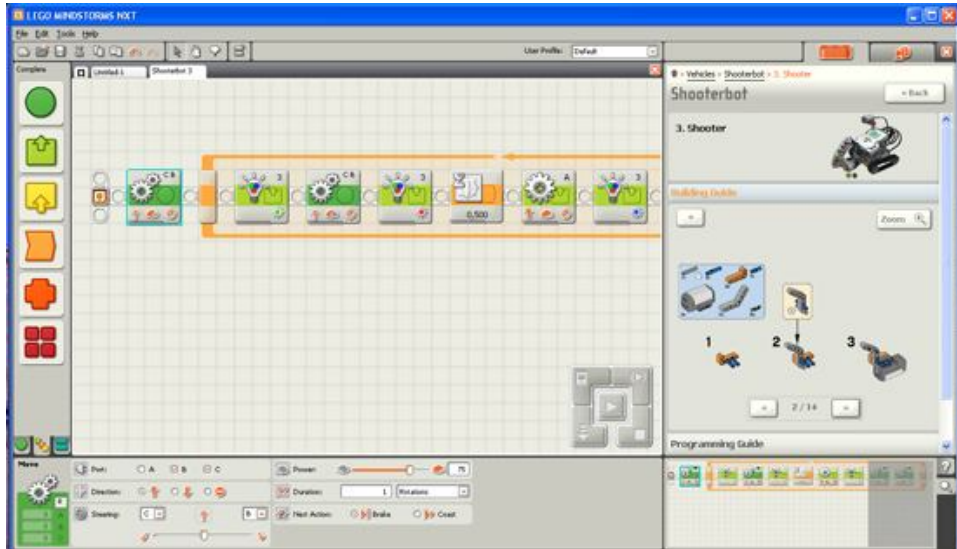


Fig. 2 – Lego Mindstorms GUI

2 BUILDING THE MANIPULATOR

Two Lego Mindstorms NXT 2.0 sets and one extension pack was used for building the manipulator. The manipulator has five degrees of freedom. Both NXT bricks and one motor form up the main platform, the arm itself is made of two parts. First motor is used for rotation of the whole arm, but it can not rotate in full angle but just in 270 degrees (because of the cables connected to other motors). There are used two motors and a big gear box for lifting the main arm upwards. Another three motors are placed in the half of the arm and they support lifting of the end of the arm, rotation of the wrist and the grip. The motors could be placed there because of the constant velocity joints and another small gearbox even though the arm is so heavy that two motors and some weight is needed for a balance. – see Fig 3.



Fig. 3 – Manipulator

This project uses two NXT Bricks, because of six servo motors. Each microcontroller is able to communicate with just three of them. On the other hand every microcontroller is able to drive the motor very accurately with many specifics like torque, driven distance and speed. It can also synchronize two motors to rotate exactly same speed for exactly same time. The last ability is usable with only two motors.

The NXT brick can not drive all three motors at the same time. It also can not drive two motors with different speed or set a different distance. The NXT brick is capable to do only one command at time, so we had to combine motors for being able to operate the robot easily and not only one by one.

There are no sensors connected to NXT brick, because they are not needed. The NXT has a decoder in every motor and it can drive the motors precisely without any other sensors and also send their position in driven degrees to the PC.

2.1 Programming in MATLAB

MATLAB commands to LEGO bricks are very simple, but there is no way how to do it continuously. There have to be some specifics in commands for motors like the port of the motor, power, tacholimit and action at tacholimit. When MATLAB sends this to NXT, NXT starts moving the motor and if a new command is sent before it finishes the previous one, the command is lost. The only way how to do some moves independently is to use two bricks and send them commands separately even though there is a risk of losing the command.

The NXT bricks were connected to PC by USB 2.0 because manipulator is not moving, so Bluetooth is not necessary.

3 GRAPHICAL USER INTERFACE

Graphical user interface is programmed in MATLAB for easier connectivity to NXT bricks.

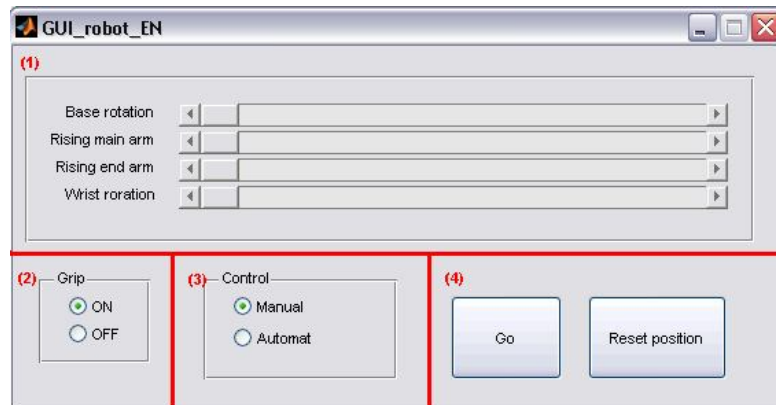


Fig. 4 – User interface

In section (1) there are four sliders which rotates and elevates with the arm. Section (2) is for control the grip, which has only two positions. Both sections (1) and (2) are active while manual control is selected in section (3). Section (3) is for choosing manual or automatic control. In section (4) there are two buttons. *Go* button is starting the movement of the arm while manual control is selected or returns the arm to starting position while automatic control is selected. *Reset* button sets new starting position.

3.1 MATLAB Code

This is part of the code in m-file in MATLAB, where are used two motors which are synchronized and on one axle.

```
% Clearing the workspace:
clear all;
close all;
COM_CloseNXT('all');

% Connection to NXTs:
h = COM_OpenNXTEx('USB', '0016530BD2E8');
l = COM_OpenNXTEx('USB', '0016530F946B');

% choosing the active one
COM_SetDefaultNXT(h);
%COM_SetDefaultNXT(l);

% Data from NXT motor
data = mUp.ReadFromNXT();
Position = data.Position;

% Data from GUI
NewPosition=700;

% Program:
Length=abs(Position-NewPosition); % Length which must be driven.
```

```

if(Length~=0) % If no change, program ends.

if(NewPosition>Position)
    DrivingSpeed=100; %speed for rising the arm
else
    DrivingSpeed=-50; %speed for lowering the arm
end %if2

% Two motors in synchronized mode:
Ports =[MOTOR_A; MOTOR_B];

Quarterturntick = 360;
mStraight = NXTMotor(Ports);
mStraight.SpeedRegulation = false;
mStraight.SmoothStart = true;
mStraight.Power = DrivingSpeed;
mStraight.TachoLimit = Length;
mStraight.ActionAtTachoLimit ='Brake';%Also can be used HoldBrake or Coast.

% Sending command to NXT:
mStraight.SendToNXT();

% Waiting for NXT:
mStraight.WaitFor();
end% If1

% Closing the comports to NXTs
COM_CloseNXT(All);

```

REFERENCES

HLINOVSKÝ, M., KRUCZEK, A.2010. Předmět ROBOTI na ČVUT FEL v Praze. *In Proceedings of Conference ARTEP'10*", Stará Lesná, 24.-26. 2 2010.Košice : Technická univerzita Košice, 2010. ISBN 978-80-553-0347-5.