The wireless communication in the walking robot application

Sylwester Cyrwus
AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail: cyrwus@student.agh.edu.pl

***Abstract*—The paper presents the identification process of time delays in the development environment created for the wireless control of a hexapod, the six-legged walking robot. RFM12B transceiver modules are used for communication between the host computer and the FPGA board. The board was designed to control all 18 servo motors. The software and hardware components are described in detail. The advantages and disadvantages of the designed communication system as well as the servo motors driver based on the FPGA circuit are listed. Various parameters of the control system are investigated. The experiments that allow statistical analyses of time delays in the communication system are described and the results are included. Thorough analysis of time delays are presented in numerical and graphical forms.**

***Index Terms*—hexapod, walking robot, identification, wireless communication, FPGA**

I. Introduction

The hexapod is a walking robot equipped with the six identical legs. Each of them consists of three servo motors. This kind of construction is a simplification of natural six-legged insects like a cockroach. Hexapods may be used to test biological theories about the insect locomotion, motor control and neurobiology. Insects are chosen for the biological tests because their nervous system is simpler than other animal species. Figure 1 shows the three-dimensional model of the hexapod created using the Autodesk Inventor 2009® software application.

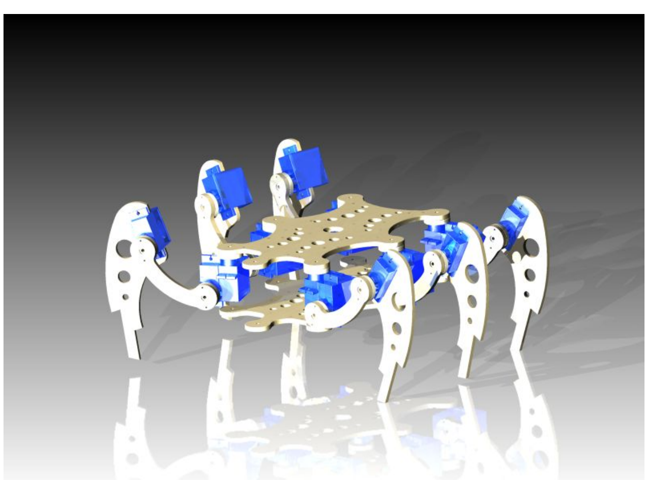

Figure 1. Model of the hexapod - six-legged application

The Turnigy TR-1160A mini servo motors, with a range of movement from about -70 to +70 degrees are used in the application. The control signal of the servo motor is a square wave, similar to the PWM signal. The width of its "high" level corresponds to the desired servo motor position. Figure 2 shows the control signal with expected time and voltage values. The position of the servo motor is measured by the built-in potentiometer. A voltage drop on the potentiometer gives direct information about the servo motor position.

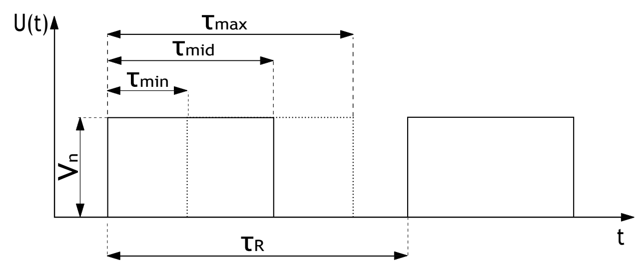

Figure 2. The servo motor control signal.
V_n- 3V - voltage value, τ_{max}- 2.5ms - the servo motor right boundary position, τ_{mid}- 1.5ms - the servo motor mid-position, τ_{min}- 0.5ms - the servo motor left boundary position, τ_R- 20ms - square pulse refresh time

The development environment requires a minimum of 18 PWM-like actuators and the ability to generate continuously the PWM-like signals. These requirements are the result of the fact that the system has to continuously control all of the robot degrees of freedom.

The identification of the time delays in the wireless communication is necessary for the correct implementation of the control algorithm and the proper design of the structure of the wireless communication.

II. The architecture of the development environment

The designed architecture of the development environment is splitted into two main layers: the hardware and the software. The hardware layer is responsible for the setting required position of each servo motor. The software layer is responsible for the algorithm implementation and the hardware configuration management.

A. The hardware layer

The hardware layer is an intermediate layer between the robot and the host computer. The FPGA circuit was applied in this part in order to be used in scientific research. This circuit gives the opportunity to control all the servo motors independently. In the current version of the environment the FPGA circuit is used to generate PWM-like signals to control the angle of the servo motors. The circuit could also be equipped with differential analog to digital converters used to the feedback control of the servo motors position. The AVR Atmega8 microcontrollers and the RFM12B transceivers are used to communicate the host computer with the FPGA circuit.

B. The software layer

The control PC programme was written in Microsoft Visual C++ 2008 Express Edition® environment and runs under the Microsoft Windows® operating system. The main advantage of the Microsoft Windows® operating system and the C++ programming language is the popularity among students. Nevertheless the Microsoft Windows® is not the hard real-time operating system what means that several time delays may appear which are unable to predict.

A host PC is used to control the movement of the robot. The AVR Atmega8 microcontroller analyses the data received from the host computer, calculates the following position of each servo motor and transfers this information to the FPGA using the RFM12B transceivers. The FPGA sets the direct position of each servo motor continuously due to the information from the host computer. The ISE WebPack and the GCC compiler were applied. For the time delay analysis MATLAB® software was used.

III. The communication

The wireless communication between the host computer and the FPGA board is conducted using the RFM12B transceiver modules and the AVR Atmega8 microcontrollers. Figure 3 shows the scheme of the development communication architecture.

The data transmission can be divided into five main parts:

a) during the first part, the host computer sends 8-bit information about the following movement of the legs using the RS-232 standard. This information includes the direction, radius and speed of movement. The parameters of the RS-232 standard are specified below:

- baud rate: 57,600
  - byte size: 8
  - stop bits: 1
  - parity: none
  - flow control: none

b) the next step is to calculate the following position of each servo motor according to the information received from the host computer. The AVR Atmega8 microcontroller analyses all the necessary data. After having finished all calculations the information about the next position of each servo motor is transmitted to the RFM12B transceiver using the SPI protocol

c) the first RFM12B transceiver sends the data to the second RFM12B transceiver. The wireless communication is conducted with the following parameters:

- band: 433 MHz
  - operation frequency: 430.8 MHz
  - base band bandwidth: 134 kHz

d) the second RFM12B transceiver sends the received data to the next AVR Atmega8 microcontroller using the SPI protocol

e) during the last part of the data transmission the AVR Atmega8 microcontroller divides all the received data into 4 bit parts and sends it to the FPGA Spartan III microcontroller.

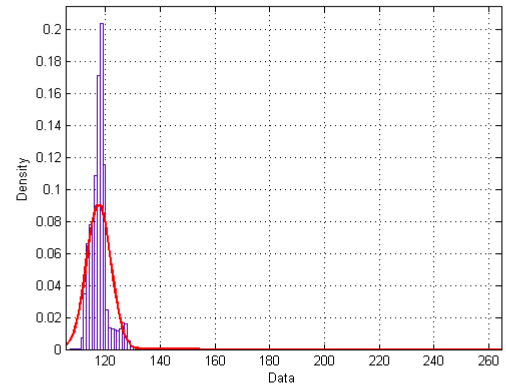

Figure 4. Density distribution of the time delays during data transmission for a distance of one metre and fifty centimetres, without any obstacles between the transmitter and the receiver.

IV. Experiments

Several experiments have beed carried out. During all of them the time delays in the wireless communication were measured with the GetSystemTimeAsFileTime function. This function is exported by kernel32.dll and retrieves the current system date and time. The information is in Coordinated Universal Time (UTC) format. Experiments were repeated for the different distances with or without such obstacles as a wooden board, or a glass or concrete wall. The analysis of the results will indicate the effect of these factors on the time of the communication.

During the first experiment the distance between the transmitter and the receiver was one metre and fifty centimetres, without any obstacles. Figure 4 presents density distribution of the time delays during the experiment. The time delay was measured 20,000 times.

In the following research the distance between the transmitter and the receiver was not changed but the receiver was placed inside the three centimetres thick wooden box. Figure 5 presents density distribution of time delays during the experiment. The time delay was measured 20,000 times.

Another 20,000 measurements were made with the same distance between the transmitter and the receiver as the previous two, but this time the transmitter was placed behind the ten centimetres thick concrete wall. Figure 6 presents density distribution of time delays during the experiment.

Two further experiments were made for the distance of ten metres between the transmitter and the receiver sequentially without any obstacle and with a ten centimetre concrete wall. The distributions of the time delays during the data transmission in these experiments are presented on figure 7 and figure 8.

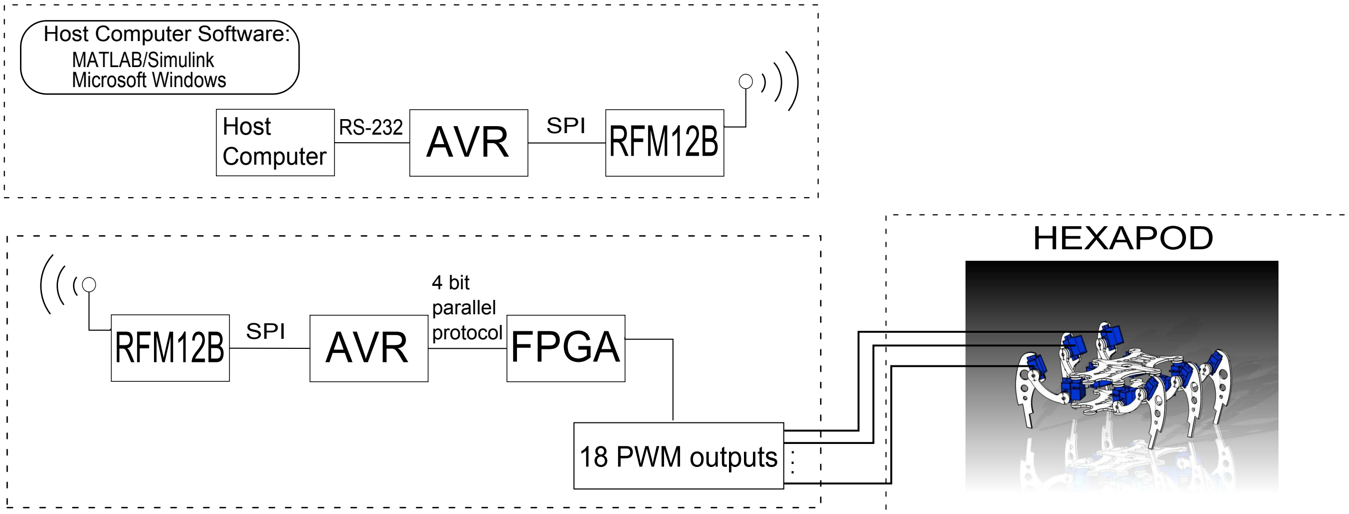

Figure 3. The scheme of the development communication architecture.

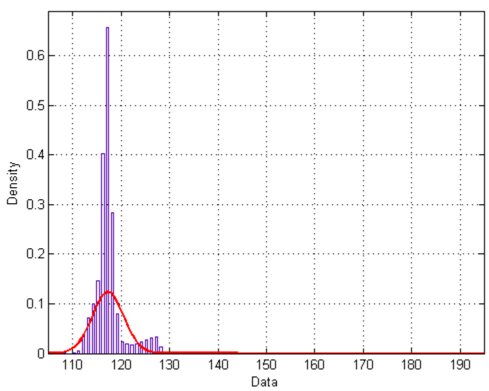

Figure 5. Density distribution of the time delays during data transmission for a distance of one metre and fifty centimetres, with the receiver closed inside the three centimetres thick wooden box.

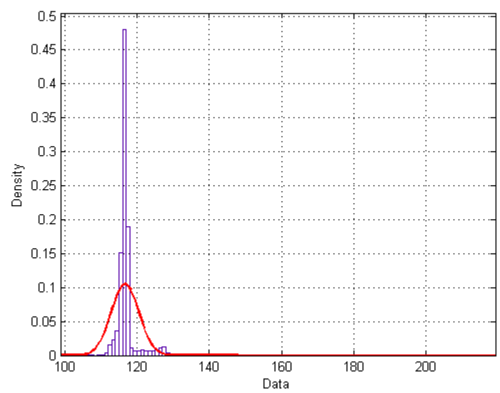

Figure 7. Density distribution of time delays during data transmission for a distance of ten metres, without any obstacles between the transmitter and the receiver.

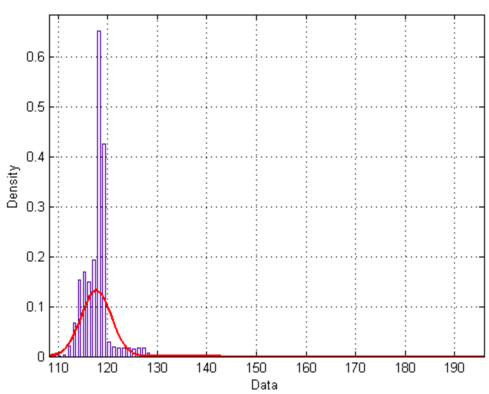

Figure 6. Density distribution of time delays during data transmission for a distance of one metre and fifty centimetres, with the transmitter placed behind the ten centimetres thick concrete wall.

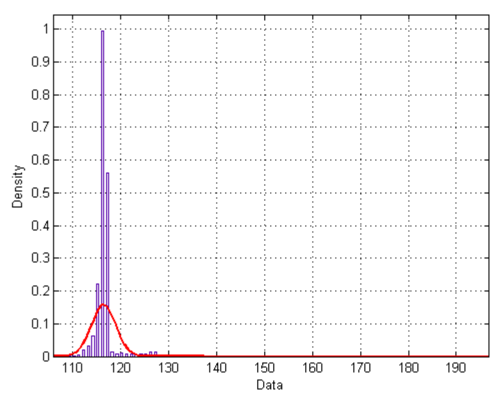

Figure 8. Density distribution of the time delays during the data transmission for the distance of ten metres, with the transmitter placed behind the ten centimetres thick concrete wall.

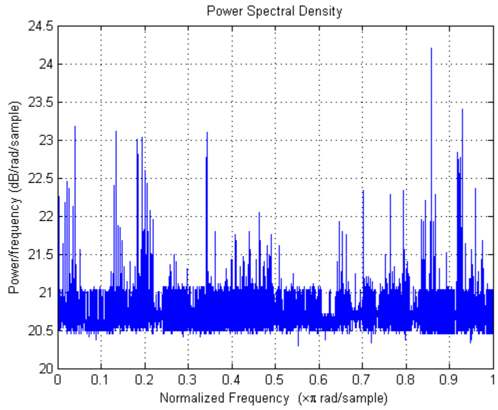

Figure 9. Spectral power density of the time delays during the data transmission for a distance of one metre and fifty centimetres, without any obstacles between the transmitter and the receiver.

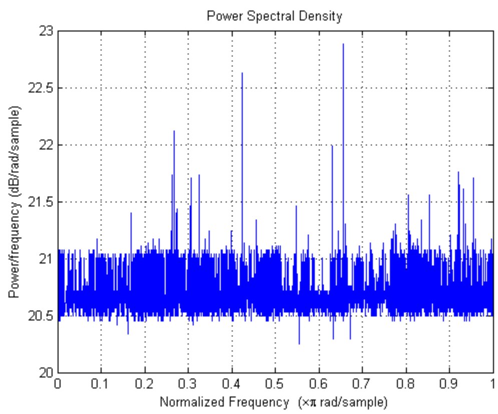

Figure 10. Spectral power density of the time delays during the data transmission for the distance of one metre and fifty centimetres, with the receiver closed inside the three centimetres thick wooden box.

The next five figures present spectral power density of the time delays measured during the experiments.

V. Summary

Taking into consideration all experiments that were carried out, it can be concluded that neither the various obstacles nor the distance between the transmitter and the receiver have the influence on the time delays in the wireless communication. The expected value of the time delays was nearly the same during all tests. For the proper modelling of the communication system there have to be included the constant delay in the form of $e^{-s\tau}$ with additional delay in the form of the random variable with the parameters such as the identified time delays.

The experiments show that the identification of the time delays in the wireless communication was necessary for the correct implementation of the control algorithm and the proper design of the structure of the wireless communication. The results indicate the need to re-design of the actual control structure. In the current version the first AVR Atmega8 microcontroller after receiving from the host PC the information about the following movement calculates the position of each servo motor and record this data in 288 bits (16 bits for each of 18 servo motors). This information has to be sent via the SPI protocol two times (from the first AVR Atmega8 microcontroller to the RFM12B transceiver and from the second RFM12B transceiver to the next AVR Atmega8 microcontroller), through the 4 bit parallel protocol from the second AVR Atmega8 microcontroller to the FPGA and wirelessly between the RFM12B transceivers. The new version of the communication architecture has to be designed to avoid the time consuming 288 bit data transmission

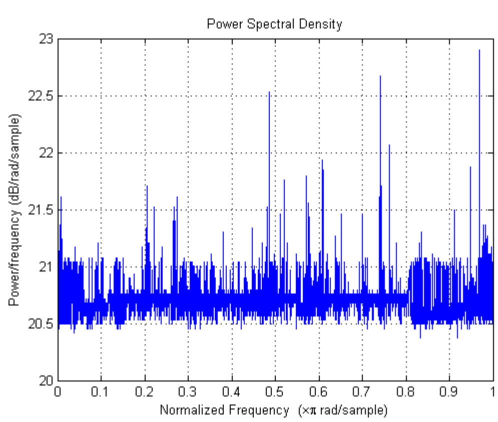

Figure 11. Spectral power density of the time delays during data transmission for a distance of one metre and fifty centimetres, with the transmitter placed behind the ten centimetres thick concrete wall.

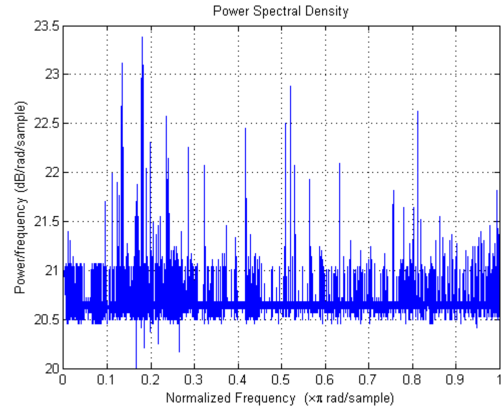

Figure 12. Spectral power density of the time delays during data transmission for a distance of ten metres, without any obstacles between the transmitter and the receiver.

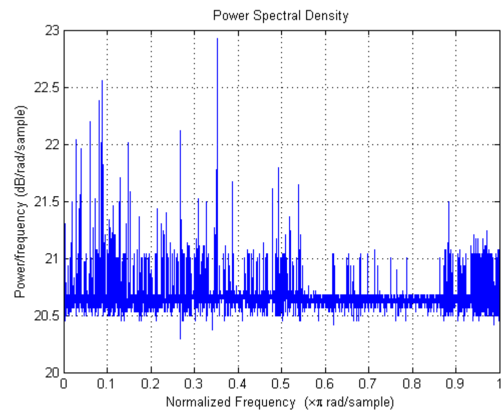

Figure 13. Spectral power density of the time delays during data transmission for a distance of ten metres with the transmitter placed behind the ten centimetre concrete wall.

Experiment	Distance [m]	Obstacle	Number of samples	Minimum time [msec]	Maximum time [msec]	Expected value [msec]
1	1.5	none	20,000	107	263	117.5089
2	1.5	3 cm wooden box	20,000	106	194	117.3003
3	1.5	10 cm concrete wall	20,000	109	195	117.6708
4	10	none	20,000	100	218	116.6832
5	10	10 cm concrete wall	20,000	107	196	116.4334

Table I
SUMMARY

from the first AVR Atmega8 microcontroller to the FPGA. In the re-designed structure the 8 bit information about the parameters of the following movement will be sent from the host PC up to the second AVR Atmega microcontroller. Only then all the calculations will be done and data will be transmitted to the FPGA. This modification will have a considerable impact on the time delays in the wireless communication.

Due to the low AVR Atmega8 microcontroller computing power the following position of each servo motor during the different movements will be recorded in the LookUpTables. After having received the information about the parameters of the following movement the AVR Atmega8 microcontroller will retrieve the relevant data form the LookUpTable and rend it.

Despite such long-time data transmission, the system is slow enough to work properly. Even the 200 ms time delays between the successive data have no appreciable effect on the control of the robot.

This tested wireless communication technology will be used to control robots from a PC, and can be applied in laboratories as well as in low cost student projects.

REFERENCES

[1] M. Piątek and P. Piątek, *Development environment for the walking robot application,* Computer Methods and Systems 2009, 295-299, November 2009, Krakow.

[2] M. Piątek, *Identification of the servo motor used in the walking robot,* Półrocznik AGH Automatyka 14(1), volume 14, 2010.

[3] Praca zbiorowa pod redakcją J. Kalisza, *Język VHDL w praktyce*, Wydawnictwa Komunikacji i Łączności, Warszawa, 2002.

[4] M.W.Spong and M.Vidyasagar, *Dynamika i sterowanie robotów,* Wydawnictwo Naukowo - Techniczne, Warszawa, 1997, transladed to Polish.

[5] J. Grębosz, *Symfonia C++*, Editions 2000, Kraków, 2008.

[6] A. Witkowski, Mikrokontrolery AVR programowanie w języku C przykłady zastosowań, Polska Akademia Nauk, Katowice, 2006.

[7] B.Mrozek, Z.Mrozek, *Matlab and Simulink, User Manual*, Helion, 2004, in Polish.

[8] T. Zielińska, *Maszyny kroczące*, Wydawnictwa Komunikacji i Łączności, Warszawa, 2005.

[9] A.Turnau, *Model identyficaion dedicated to the time-optional control*, In 17th IFAC World Congress, 2655-2660, July 2008.