Hexapod Robot: Test Platform for Bio-Inspired Controllers

Marek Zak, Jaroslav Rozman, Frantisek V. Zboril
Faculty of Information Technology
Brno University of Technology
Brno 612 66, Czech Republic
Email: {izakmarek, rozmanj, zboril}@fit.vutbr.cz

***Abstract*—This paper aims on design, construction and control of a hexapod robot, which is a six-legged walking robot. The result of this project is a legged robot equipped with sonars, force-sensitive resistors and encoders, which will be used as a test platform for hexapod robot bio-inspired controllers. A graphical user interface was developed to control and display data from sensors. There are also described basic characteristics of walking robots and several approaches of hexapod bio-inspired controllers.**

***Index Terms*—Hexapod; Walking Robot; Hexapod Controller**

I. Introduction

This paper dealt with the design, construction and control of hexapod robot – a test platform for bio-inspired controllers. The first part of this paper describes the basic characteristics of walking robots. The second part mentions several existing walking robots. The third part focuses on short overview of several approaches of controllers for hexapod robots. The last part is about a hexapod robot of our design, its construction and control.

If we want to design a hexapod robot, we should keep in mind several things. The most importatnt is the number of degrees of freedom. At least two degrees of freedom are needed to design and construct a functional walking robot – the first degree of freedom for lifting the leg, second degree of freedom for rotating the leg. Nevertheless, there should be three degrees of freedom for a good functioning chassis, because the legs move along a circle and the forward movement of the body causes slipping between the foot and the terrain, which can be compensated by third joint [1]. Inverse kinematics can be used to calculate the angles of each joint [2].

A. Gaits

Walking gait refers to the locomotion achieved through the movement of robot legs. Compared to human gait, the legged chassis usually has more than two legs – six in the case of hexapod robot. Therefore, the locomotion of a robot is much more complicated. There are several basic gaits, such as tripod, wave or ripple (Figure 1).

Tripod gait is based on two groups of legs. During each step the first group of the legs is lifted and is rotated forward and is laid upon on the ground. Then the other group is lifted. Now both groups are moving, the first group backward, the second group forward and finally the second group is laid on the ground. It is obvious that both groups perform the same movement, but they are shifted by half a period. Tripod gait is very fast, but also very unstable. That is because at one moment half of the whole weight of the robot is only on one leg, which can lead to slip or even to fall.

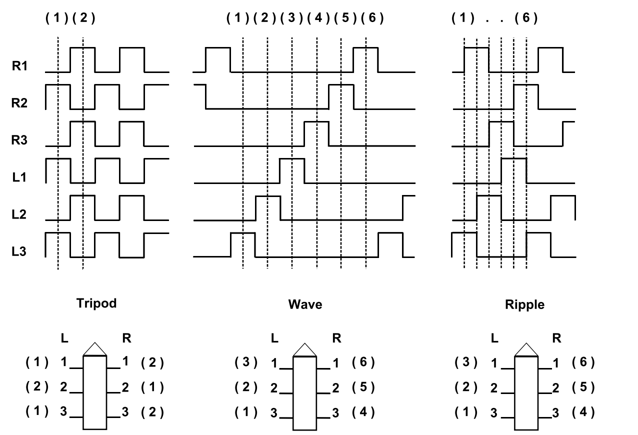

Fig. 1. Gaits. The graph shows the movement of each leg in time. A high value represents leg movement (swing phase), low values means no movement (stance phase). Tripod, wave and ripple gaits are shown in this figure. Tripod has two group of legs, all the legs in the same group move at once. This is the fastest gait for hexapod, but the least stable. In the wave gait only one leg is moving forward at any time. This leads to extremely stable movement, which is good for rough terrain. After all legs are set up to their new positions, the step is completed and body moves forward. In the ripple gait all legs move the same way, but their moves are shifted and the swing phases of neighbour legs are overlapped. Once the first leg is being placed on the ground, the second leg is already rising up. Inspired by http://www.oricomtech.com/projects/cynthia2.gif, 28. 12. 2015.

Another gait is wave, which is the most stable gait, but also the slowest. Wave gait consists of a sequential adjustment of legs forward and only when all the legs are set to the new positions, the step is completed. In each phase of step maximally one leg is lifted up, which leads to high stability of this gait and makes the wave gait ideal for a movement in rough terrain.

Ripple gait is inspired by insects. Each leg performs the same move – up, forward, down, backward. Leg moves partially overlap. In other words, the time when the first foot is lifted and begins to move forward, the second leg begins to

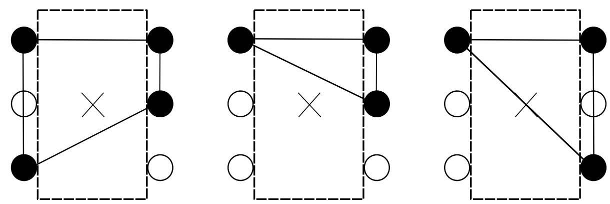

Fig. 2. Possible positions of the chassis during its movement. This figure is taken from [2].

lift up. In this way the robot cycles through all legs. This gait is quite similar to the wave gait.

There are also other common gaits such as tetrapod or rotation. These are the most common gaits. But, theoretical number of different gaits N can be calculated using Equation 1.

$$N = (2K - 1)! \quad (1)$$

where K is the number of legs [3]. It is obvious that the number of different algorithms grows rapidly with the number of legs (for robot with six legs there are 11! = 39,916,800 possible sequences of movement – gaits). However most of them cannot be used in practice, because they do not lead to efficient movement in the desired direction or cause instability and crashes of the robot. For example moving one leg up and down is one of the gaits, but has no effect on the robot movement. Still, the number of all possible gaits is quite high and it is not possible to check them all. Therefore, new approaches of automatic generating of gaits are searched for.

Walking chassis movements can be divided into statically stable and dynamically stable [2], [4] according to the used gait. Static stability represents the ability of the chassis to remain in a stable position in every moment of the movement. Static stability is typical for hexapod, which is always stable during its movement – at least three legs are on the ground. Dynamically stable chassis is sometimes out of balance – balancing or falling. This is common for two-legged robots and can be compensated by the movement dynamic. The possible positions of chassis represents Figure 2.

II. Existing Robots

Although legged chassis are not common and their use is quite limited, even today, we can find several examples, which are used in extreme conditions. An example might be ATHLETE [5], a robot by NASA (Figure 3). This six-legged robot was designed for exploration of planets, especially of Mars. His legs are equipped with wheels and it is able to walk and ride. In the field, where driving on the wheels is not possible, come the legs. The robot is also able to grab a tool and drill into the ground or carry burdens.

Another example may be LS3 [6], a robot manufactured by Boston Dynamics (Figure 4). It is a four-legged robot that is capable of reaching speeds of up 10 km/h and will serve the military for carrying material and equipment. This robot, unlike the ATHLETE, moves dynamically. That means he can stay in balance even when he has lifted two or more legs.

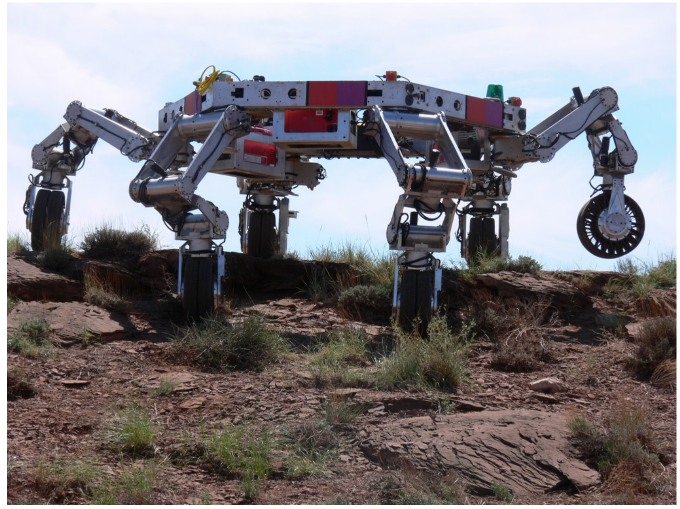

Fig. 3. ATHLETE. This six-legged robot was designed by NASA for exploration of planets, especially of Mars [5].

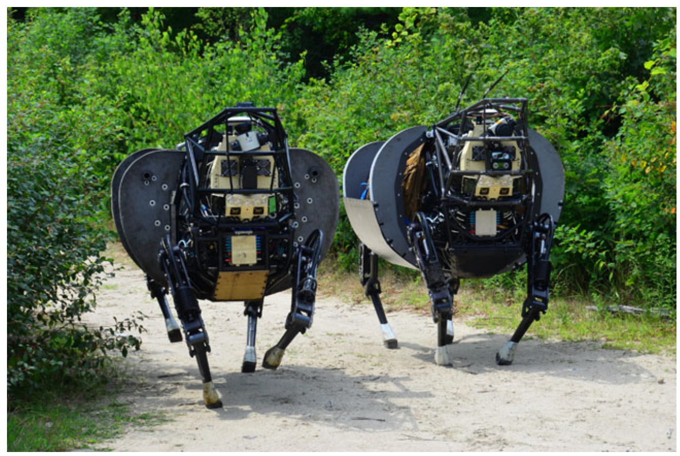

Fig. 4. LS3. A four-legged robot developed by Boston Dynamics for military purposes. The robot can reach the speed of 10 km/h [6].

The ATHLETE and LS3 are examples of working prototypes, which are designed for some specific tasks. There are also a lot of smaller robots, which were developed for research and experiments. Many of them are described in [7]. There are also described issues of designing a hexapod robot such as body types, actuators or robot proportions.

III. Bio-Inspired Controllers

Almost all approaches used in design and control of walking robots are inspired by the nature. It is not a coincidence, that all walking robots with four and more legs looks like some animals. The construction of their body is well formed and verified through a long evolution. But we can find inspiration not only when building a walking robot, but also when controlling it. Except of evolution methods, which are also

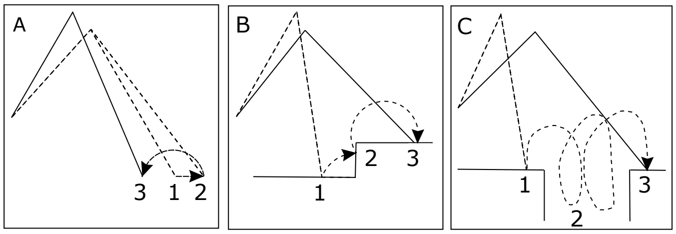

Fig. 5. A) Stepping reflex. The leg can step from the position (2) to the position (3) to better support the body. B) Elevator reflex. If the leg encounters an obstacle (2), it tries to lift the leg higher to step over the obstacle. C) Searching behaviour. If the leg cannot reach ground at expected location (2), it tries to find another foothold (3). This figure is taken from [11].

inspired by nature, we can study, how animals solve difficult situation during their movement.

Beer et al. [8] discuss bio-inspired robots and controllers. They point out, that distributed controllers are more suitable for locomotion generation than controllers with one centralized system. This is similar to the insect approach. Espenschied and Quinn [9] describe a bio-inspired hexapod robot. Its controller was firstly developed in simulated environment and then applied to real hexapod robot. This robot is more insect-like than its predecessor in the terms of leg configuration and degrees of freedom. The robot is capable of turning, walking on a rough terrain and walking quickly.

Studies of animal nervous system show that the pattern of locomotion is controlled by neural centers known as central pattern generators (CPGs), whose output is an oscillating signal with a certain frequency. These CPGs are also widely used to generate control signals for walking chassis. Ijspeert et al. [10] present a spinal cord model. They address three fundamental issues related to vertebrate locomotion: the modifications of the spinal locomotor circuits during the evolutionary transition from aquatic to terrestrial locomotion, the mechanisms necessary for coordination of legs, and the mechanisms of gait transitions. They create a CPG model, which is composed of a body CPG and a leg CPG implemented as a system of coupled nonlinear oscillators. The CPG model produces walking and swimming patterns, which are similar to the real salamander patterns. It was observed in stimulation experiments of mesencephalic locomotor region, that the model produces transition between gaits by changing the drive. The swimming and the walking movement of the robot is similar to real salamander.

Espenschied et. al. [11] proposed using reflexes, which were observed in insect. When the leg moves forward, stepping, elevator and searching reflexes are used to find suitable position for the leg. See Figure 5.

The stepping reflex ensures, that the robot keeps the legs in the best positions to spare energy or to better support the body. If it is possible, the leg is moved closer to the body.

The elevator reflex is used when the leg is moving to new position. If the leg encounters an obstacle and cannot finish its move, it tries to lift the leg higher and step over the obstacle.

Searching reflex is used when the leg cannot reach the ground at expected location. It then tries to find another foothold to support the body and finish the step.

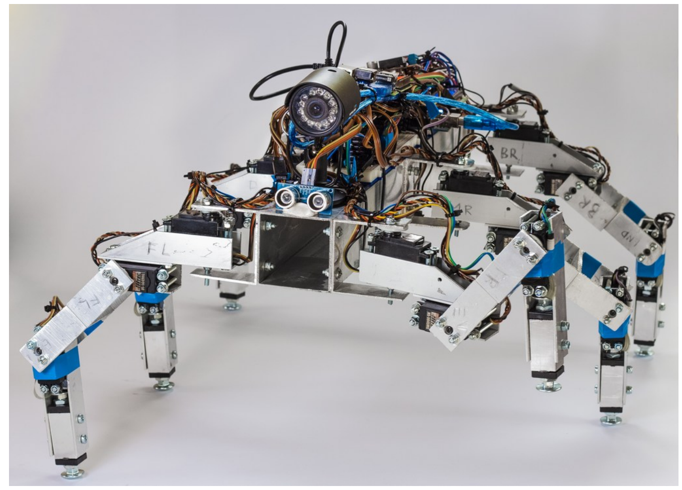

Fig. 6. Hexapod. This robot was designed and constructed during the project. It is equipped with sonars, ground sensors, camera, LCD display and more accessories.

Ferrell [12] compares three different insect-inspired locomotion controllers – reflexive, hybrid and patterned. Each controller was tested while unloaded (walking while suspended above ground), loaded (walking on flat terrain), with lesion (loss of a leg) and with external leg perturbations.

IV. Robot Body Construction

We constructed a hexapod robot (Figure 6) during this project. This hexapod has rectangular body type – it has two groups of legs, three on each side. Each leg has three degrees of freedom and is powered by hobby servomotors HS-5485HB on coxa and tibia joints and HS-5645MG on femur joint (Figure 7). Servomotors are equipped with encoders and leg bases are equipped with force-sensitive sensors to detect ground (Figure 8).

Servomotors must be sufficiently powerful, depending on the desired behavior. If tripod gait is required, then each motor on the middle legs must be powerful enough to hold half of the weight of the robot. If the robot has to carry loads, it has to be added additional power on each motor.

It is important to choose a suitable material for the body. It must be solid enough, but not too heavy. Therefore aluminum profiles were chosen for the body construction. They are quite light-weighted and solid enough. In addition, they are available in various sizes and shapes and they are easy to handle. Robot is made of 25 mm and 60 mm profiles. Structure of the leg is shown in Figure 7. The robot is 70 cm long, 47 cm wide and 6 cm high when lying on the ground, however these dimensions are different during the walk. The robot weighs 4.3 kg.

V. Robot Electronic System

In order to control the robot there must be some control unit. The open-source electronic platform Arduino Mega 2560

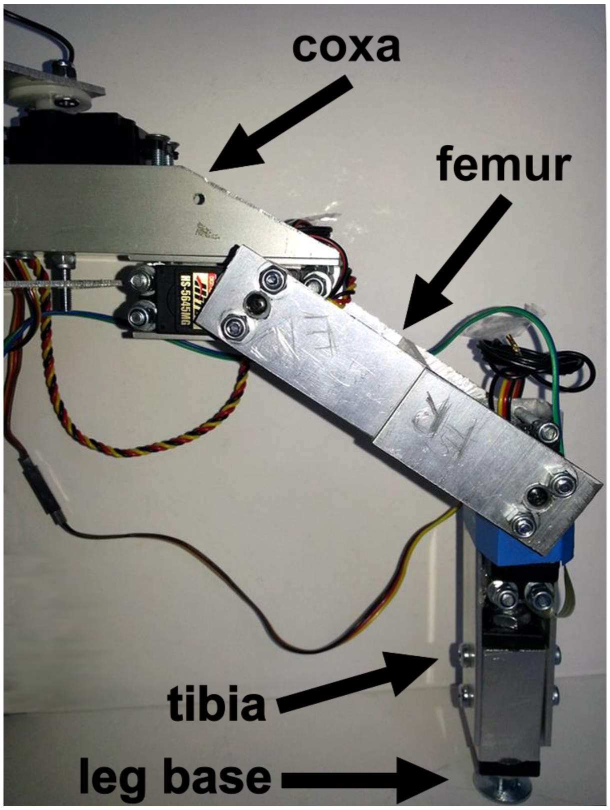

Fig. 7. Leg structure. From left: body is connected to the coxa using coxa joint, which allows forward and backward rotation. Coxa is then connected to the femur and femur is connected to the tibia, which allows lifting and laying. [1]

Fig. 8. Force-sensitive resistors placement. The inner profile is movable in digits of singles a millimeter. This ensures perfect functionality of the force-sensitive resistor.

[13], which is based on MCU Atmega2560 [14], was chosen to drive the servomotors and sensors on this robot. Arduino board is connected to the Raspberry Pi [15] via USB cable. The whole scheme is in Figure 9. All sensors like sonars, LCD display, memory card, GPS module or force-sensitive resistors are connected to Arduino. LCD display is connected by digital pins using integrated Hitachi HD44780 driver, which allows 4-bit or 8-bit mode. The 4-bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. There are also 18 servomotors connected to digital pins and driven by MCU's timers.

Raspberry Pi is a miniature computer the size of a credit card, to which a standard monitor, a keyboard and a mouse can be connected. It has extremely low power consumption (max. 3.5 W) and can run linux based operating system Raspbian. There are several models, which differ in RAM, the number of USB ports or GPIO pins. Our robot is eqquiped with Raspberry Pi model B+, which is equipped with a USB Wi-Fi dongle providing connection to a wireless network, and runs Qt [16] client program, which is able to find the IP address of the server and gets connected to it. Sensor data are sent to a computer after successful connection. Client is also capable of reconnection after disconnection.

All servomotors and sensors are connected and driven by the Arduino board. Servomotors are controlled by MCU's timer using VarSpeedServo library [17].

Robot is also equipped with ultrasonic distance sensors HC-SR04 – sonars. These sensors can measure distance from 2 cm to 400 cm. It has 4 pins – Vcc, ground, trigger and echo. Robot is equipped with eight sonars, three in front, three in back and two on sides. Sonars are connected using MCU's digital pins and a multiplexer.

If the robot should walk across rugged terrain, it must be able to detect ground. Therefore each leg is equipped with force-sensitive resistors, which are better than tactile switches, because they can detect level of pressure on the leg. This is used for robot calibration. To determine the pressure force-sensitive resistor is connected to A/D converter using voltage divider. The values form ground sensors are also used to evenly distribute the robot weight on all legs.

The robot also has a GPS module to determine its position and a SD card to store logs and configuration files.

Besides of the sensors, the robot is equipped with a two-line LCD display, which displays basic information about the robot such as battery level, the name or the selected gait.

Energy to the entire system is supplied by one 11.1 V Li-Po battery, which can supply up to 60 A. This power is sufficient for all servomotors and electronics. But servomotors require 6 V power supply and electronics require 5 V power supply. Therefore, the robot is equipped with voltage regulators. Each servomotor has its own regulator and there is one more regulator for electronics. Switching voltage regulator were used to decrease power consumption. The robot can operate about two hour on one battery. To extend this time robot can be equipped with second battery, which doubles the operation time.

VI. Robot Control Software

The software of the robot is divided into three parts – a computer user interface program writen in C++ and Qt, a Raspberry Pi program also writen in C++ and Qt and an Arduino C++ program. Data from Arduino are sent to the Raspberry Pi through a USB serial port and then to a computer over Wi-Fi.

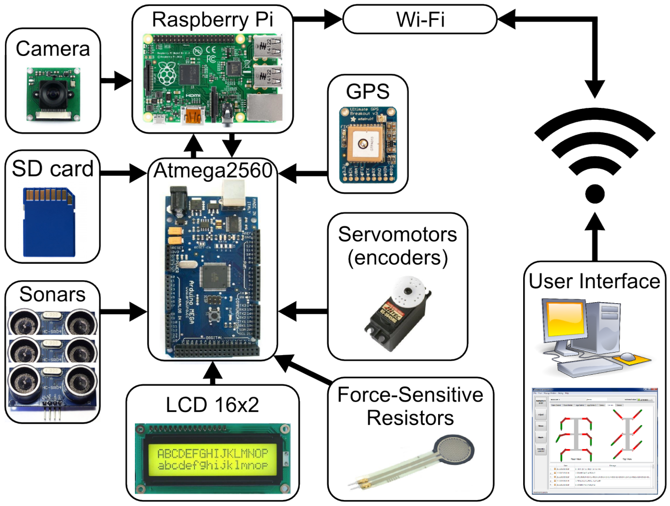

Fig. 9. The electronic system of the robot. In the center is a MCU Atmega2560 integrated on a Arduino Mega 2560. Most of the sensors like sonars, LCD display, memory card, GPS module or force-sensitive resistors are connected to it. There are also 18 servomotors connected to digital pins and driven by MCU timers. Arduino board is connected to the Raspberry Pi via USB cable. Raspberry Pi is connected to the computer via wi-fi and provides data from the sensors to the computer and commands from the computer to the Arduino. Data from the sensors are visualized in the user interface on the computer.

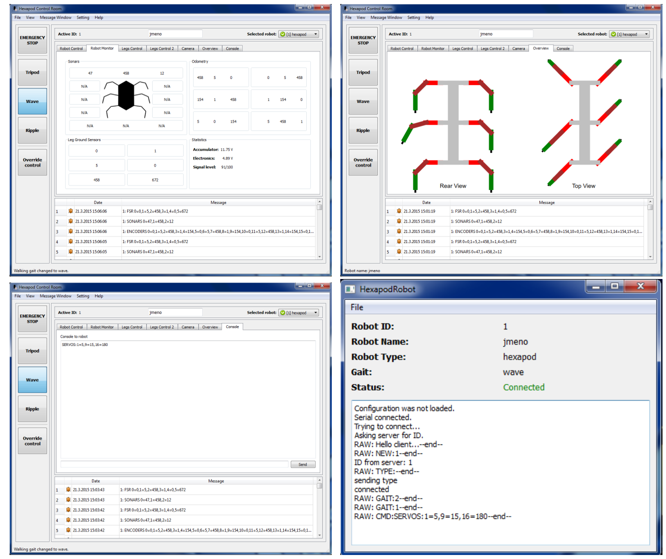

Fig. 10. User interface screenshots of Hexapod Control Room. Top left figure shows the Monitor screen with the values of sonars, encoders and force-sensitive resistors. There is also information about battery level and robot wi-fi signal strength. Top right image shows the actual position of the legs. Left model is from back view and right model is from top view. Bottom left image presents console, which allows sending command to the robot. Bottom right image shows the client window with the most important information and log window.

A. Hexapod Control Room

Hexapod Control Room is user interface program (UI) (Figure 10) designed in C++ and Qt [16], which is used to control servomotors, gaits and monitor robot sensors. It also shows the position of the robot legs from the top view and from the back view and provides command and log windows. The server allows up to 10 connections and it is possible to switch among the connected robots. Inactive robots are automatically removed or asked to reconnect.

There was also developed Qt library for custom and external applications. It allows to create a server, which creates connection to the client and returns the raw data from the client. This is useful when developing custom control application for the hexapod robot.

B. Robot Client Program

This program runs on Raspberry Pi and communicates with Hexapod Control Room and Arduino MCU. User interface is shown in Figure 10. After the program starts, it tries to connect to the server. The IP address is discovered using broadcast messages and the client asks for a new identifier and sends robot type and eventually its name. After exchange of these messages the new robot is added into the server user interface and the client starts sending data from sensors, which are displayed in the Hexapod Control Room. For initial connection TCP sockets were used to prevent data lost, however, sensor data are sent through UDP sockets, which have higher performance and allow real-time communication.

A console version of client program was developed to decrease robot power consumption and to spare Raspberry Pi CPU performance. This program lacks user interface and run only as a console application, however it has better performance.

C. Microcontroller Program

The MCU on Arduino runs program, which communicates with the Raspberry Pi over USB serial port. Once the MCU is authorized from the Raspberry Pi, the program starts sending data from sensors over serial port. It also controls all servomotors, manages sonar measurements and receives commands from the computer.

It is also important to keep the main program loop non-blocking. Therefore individual steps are performed according to the calendar structure, which is similar to the next-event algorithm. The steps are planned and stored in the calendar structure according to their start times. When performing a step, first item from the calendar is selected and executed. So the non-blocking execution of the main loop is achieved, which allows other events to run.

The robot is equipped by ultrasonic sonars to detect obstacles in the environment around the robot. If distance measurement is needed, a short pulse is emitted to the trigger pin, which causes the formation of several ultrasonic waves. Waves are reflected in the environment and are detected by the receiver. Based on the time elapsed from sending waves the distance can be calculated. Because measuring the time using Arduino pulseIn function [18] generates delay, interrupts were used. The time measurement is started by the arrival of rising signal and ended by falling signal. The distance can be calculated according to the time between these two events.

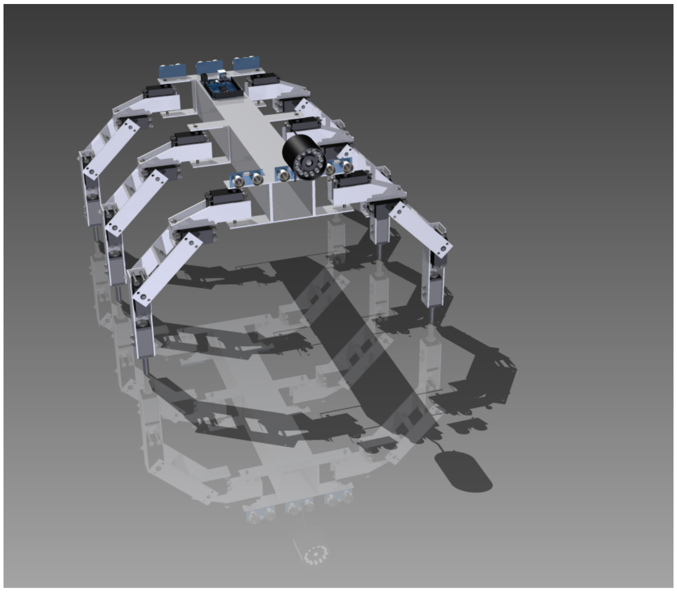

Fig. 11. The 3D model of the robot.

VII. Conclusion

This paper dealt with a hexapod robot – a testing platform for bio-inspired controllers. A short overview was given about the bio-inspired controllers. New researches show, that central pattern generators are very suitable to generate control signals for legged robots. We have placed several characteristics of legged robot and their most common gaits. We also mentioned several existing walking robots and we introduced a hexapod robot of our design, which will serve as a testing platform for our controllers.

Currently we are working on a simulation environment. An exact 3D model (Figure 11) of our robot was imported into V-REP simulation software and it will be connected to the Hexapod Control Room software. In the future we will focus on the design of the controller.

Acknowledgment

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), from IT4Innovations excellence in science project (IT4I XS - LQ1602) and by the project IGA FIT-S-14-2486.

References

[1] X. Ding, A. Rovetta, J. Zhu, and Z. Wang, *Locomotion analysis of hexapod robot*. INTECH Open Access Publisher, 2010.

[2] S. Manoiu-Olaru, M. Nitulescu, and V. Stoian, “Hexapod robot. mathematical support for modeling and control,” in *System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on*, Oct 2011, pp. 1–6.

[3] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, 3 2004.

[4] U. Saranlı, “Dynamic locomotion with a hexapod robot,” Ph.D. dissertation, The University of Michigan, 2002.

[5] NASA, “All-terrain hex-limbed extra-terrestrial explorer,” http://athlete.jpl.nasa.gov/, 2009, [Online; visited 30-09-2015].

[6] Boston Dynamics, “Boston dynamics: Dedicated to the science and art of how things move.” http://www.bostondynamics.com/robot_ls3.html, 2015, [Online; visited 30-09-2015].

[7] F. Tedeschi and G. Carbone, “Design issues for hexapod walking robots,” *Robotics*, vol. 3, no. 2, pp. 181–206, 2014.

[8] R. D. Beer, R. D. Quinn, H. J. Chiel, and R. E. Ritzmann, “Biologically inspired approaches to robotics: What can we learn from insects?” *Communications of the ACM*, vol. 40, no. 3, pp. 30–38, 1997.

[9] K. S. Espenschied and R. D. Quinn, “Biologically-inspired hexapod robot design and simulation,” in *Conference on Intelligent Robotics in Field, Factory, Service, and Space(CIRFFSS’94), Houston, TX*, 1994, pp. 21–28.

[10] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, “From swimming to walking with a salamander robot driven by a spinal cord model,” *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[11] K. S. Espenschied, R. D. Quinn, R. D. Beer, and H. J. Chiel, “Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot,” *Robotics and autonomous systems*, vol. 18, no. 1, pp. 59–64, 1996.

[12] C. Ferrell, “A comparison of three insect-inspired locomotion controllers,” *Robotics and Autonomous Systems*, vol. 16, no. 24, pp. 135 – 159, 1995, moving the Frontiers between Robotics and Biology. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0921889095001476

[13] Arduino, “Arduino - arduinoboardmega2560,” http://arduino.cc/en/Main/ArduinoBoardMega2560, 2015, [Online; visited 21-03-2015].

[14] Atmel, “Atmega2560,” http://www.atmel.com/Images/Atmel25498-bitAVRMicrocontrollerATmega640128012812560256l_datasheet.pdf, 2015, [Online; visited 30-09-2015].

[15] R. P. Foundation, “Respberry pi,” http://www.raspberrypi.org/, 2014, [Online; visited 09-04-2015].

[16] Digia, “Qt project,” http://qt-project.org/, 2015, [Online; visited 21-03-2015].

[17] N. T. Group, “Varspeedservo,” https://github.com/netlabtoolkit/VarSpeedServo, 2013, [Online; visited 21-03-2015].

[18] Arduino, “Arduino - pulsein,” http://arduino.cc/en/Reference/pulseIn, 2015, [Online; visited 21-03-2015].