International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 8 (2016) pp. 125–134

Overview of Bio-Inspired Control Mechanisms for Hexapod Robot

Marek Žák[1], Jaroslav Rozman[2] and František V. Zbořil[3]

[1]Faculty of Information Technology
Brno University of Technology
Brno 612 66, Czech Republic
izakmarek@fit.vutbr.cz

[2]Faculty of Information Technology
Brno University of Technology
Brno 612 66, Czech Republic
rozmanj@fit.vutbr.cz

[3]Faculty of Information Technology
Brno University of Technology
Brno 612 66, Czech Republic
zboril@fit.vutbr.cz

Abstract: **This paper deals with overview of possible control mechanisms for hexapod robot. Basic characteristics of legged robots, a few existing robots and their pros and cons are described in the introduction of this paper. Main part of the paper is focused on the previous work done in the field of hexapod robot control, especially the usage of evolution techniques like neural networks and genetic algorithms. The last part of this paper is about a hexapod robot of our design.**
Keywords: Hexapod Robot, Walking Robot, Hexapod Control, Hexapod Controllers, Evolution Techniques for Hexapod Control

I. Introduction

Walking robots have been the subject of many researches and studies for a long time. Although they are not commonly used, their use is not negligible. Walking robots are suitable for rough terrains. They are capable to cross large holes and they can operate even after losing a leg [1]. But their control is quite difficult, because they have a lot of degrees of freedom (DoF).
When we design a controller for legged robots, we want to achieve rhythmic and fluent movement, which we can observe in animals. Insects, for example, can walk very fast over rough terrain, while changing gaits and adapting to changes in load or leg damage. But we encounter some issues. The first is, that we do not have enough knowledge about animal's neural system. The second one is, that muscles of animals are much more stronger and lighter than any motor or system that humans invented.
There is still intensive research on the field of neurobiology and engineering, so we can build robots, which will move like animals.

A. Insect Locomotion

Insect have six legs, which give them clear stability advantages over four legged animals. For this reason insect have been studied extensively and also have been used as models for the design of six-legged walking robots [2]. The insect can adapt to a loss of one or even two of their six legs without much apparent loss of performance [3].
Slowly walking insects use a wave gait (also known as a metachronal wave gait). When they walk faster they use a tripod gait. While tripod gait is the fastest, wave gait is the most stable gait. More about gaits is described in Chapter III-B. Insects can walk on irregular terrain; some insects can even walk upside down (for example on a ceiling). The designing and controlling of six legged robots – hexapods – are inspired in insects.

II. Examples of Legged Robots

Although the use of walking robots is not common even today, we can find several examples, which are used in extreme conditions. An example might be ATHLETE [4], a NASA robot. 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. The robot is shown in Figure 1.
Another example may be LS3 [5], a robot manufactured by Boston Dynamics. 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. The robot

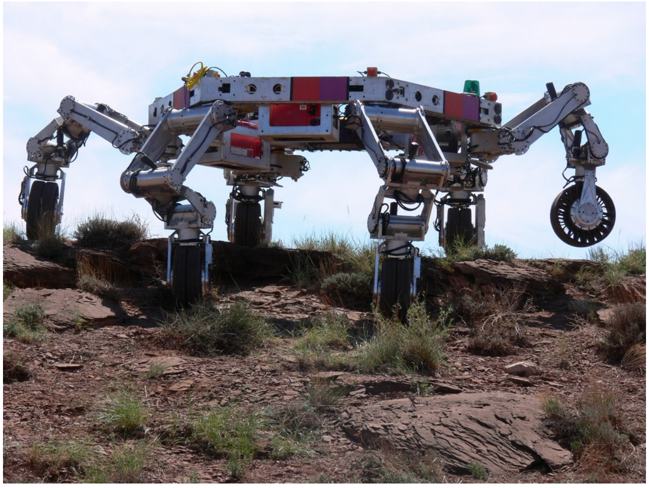


Figure. 1: ATHLETE. This six-legged robot was designed by NASA for exploration of planets, especially of Mars [4].

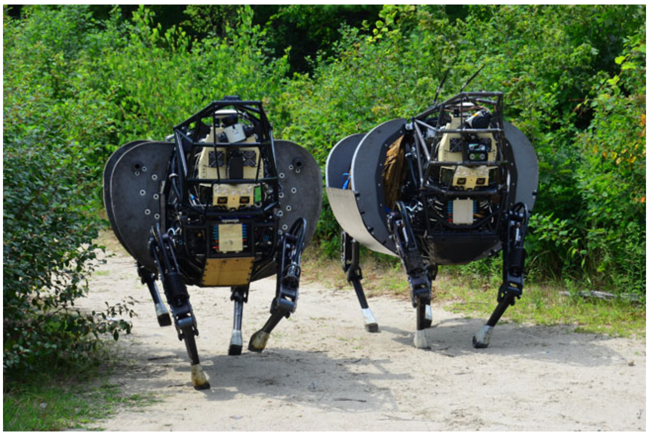


Figure. 2: LS3. A four-legged robot developed by Boston Dynamics for military purposes. The robot can reach the speed of 10 km/h.

is shown in Figure 2.
A bit different six-legged robot is RHex [6]. A number of US universities have participated on the development of this robot. Although RHex has legs, it has only one degree of freedom per leg and therefore it differs from common legged robots. However, this approach appears as very successful for the movement of the robot. RHex can travel across uneven terrain without much difficulties. The robot moves exceeding five body lengths per second (2.7 m/s), climbs slopes exceeding 45 degrees, swims, and climbs stairs. The robot is shown in Figure 3.
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. Characteristics of Legged Robots

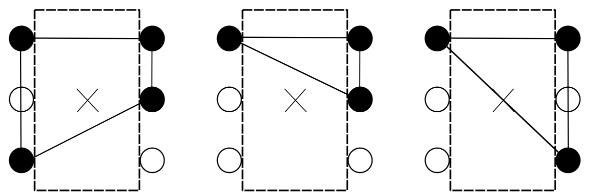
This part of the paper focuses on several characteristics of walking robots. Some classifications of walking robots and

Figure. 3: RHex. A six-legged robot with only one degree of freedom. The robot moves exceeding five body lengths per second (2.7 m/s), climbs slopes exceeding 45 degrees, swims, and climbs stairs.

most common walking gaits are described.
There are many ways how to classify walking robots - by a body shape [8], number of legs, number of degrees of freedom per leg or locomotion technique. Various options can be combined to achieve many different configurations. At least two degrees of freedom are needed to construct a walking robot - the first for lifting the leg, second 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 the third joint [9, 10].
Walking chassis can appear in three basic states during its movement based on the number of legs and the performed gait [10]. The first state is statically stable, when the chassis rests on at least three legs and is in balance. This is usual for the chassis with more legs (e.g., hexapod), which is characterized by statically stable walking (the chassis at each moment occurs in a stable position). It can also be in statically unstable state when the chassis is not balanced, which leads to collapse. This instability can be compensated with dynamic move. Then we talk about dynamically stable walking, which is a typical example of bipedal chassis. The last state is between the previous two. This is a critically stable state when the chassis balances on the edge of its center of gravity. The three states can be seen in Figure 4. These features should be also considered when designing a control system for a legged robots.

A. *Hexapod robots*

Hexapod robots are six legged robots and they belong to the group of joint leg walking robots [11]. The legs of a hexapod robot with a rectangular body are usually symmetrically distributed into two groups. Each group is located on the opposite side of the body. Another type are hexapod robots with the circle body, which have the legs evenly distributed around the body. These robots have not the front nor the rear part. In the comparison with four legged robots, hexapod robots



(a) Statically stable (b) Statically unstable (c) Critical stable

Figure. 4: Possible positions of the chassis during its movement. This figure is taken from [10].

have more redundancy because of the higher number of legs and can be theoretically move more flexible over rough terrain and can continue in movement after a loss of up to two (in some cases three) legs.

B. Walking Gaits

Management of a legged chassis is significantly more difficult than managing wheeled or tracked chassis. Just because of the fact that the wheeled or tracked chassis is able to stand on the spot due to its construction, while legged chassis need control, even when they are not moving. Also forward movement is significantly more complicated. It is not enough to activate the engine and let it run. The legged chassis needs putting legs in appropriate order.

A gait refers to the locomotion achieved through the movement of robot legs. Compared to humans, the legged chassis usually has more than two legs. Therefore, the locomotion of a robot is much more complicated. There are several basic gaits, such as tripod, wave or ripple. These gaits were observed in insects and they are shown in Figure 5.

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 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.

Another gait is wave (also known as a metachronal wave gait), which is the most stable gait, but also the slowest. Wave gait consists of a sequential adjustment of the robot legs forward. Once all the legs are set to the new positions, the step is completed. Maximally one leg is lifted up in each phase of a step. This leads to high stability of this gait.

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 lift up. In this way the robot cycles through all legs.

There are other common gaits such as tetrapod or rotation. The number of possible combinations can be expressed according to the number of legs of the chassis as:

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

where K is the number of legs [12]. It is obvious that the

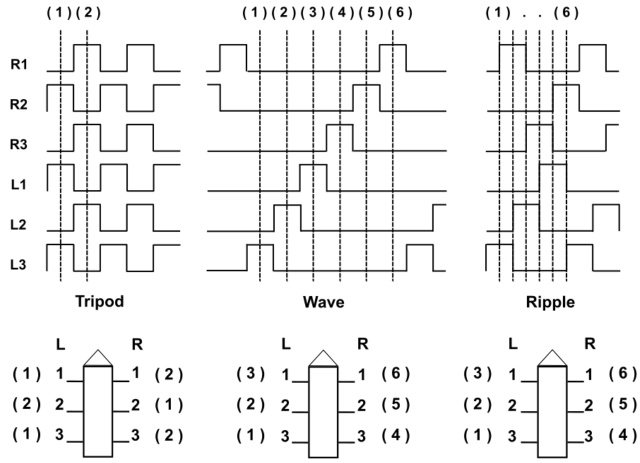


Figure. 5: Walking gaits. The chart shows the movement of each leg in time. A high value represents leg movement, low values means no movement. Tripod, wave and ripple gaits are shown in this figure. Tripod has two group of legs, all the legs in the same group move at once. In the wave gait only one leg is moving forward at any time. After all legs are set up to their new positions, step is completed. In the ripple gait all legs move the same way, but their moves are shifted. Inspired by http://www.oricomtech.com/projects/cynthia2.gif, 30. 9. 2015.

number of different gaits 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 or cause instability and crashes of the robot. Still, the number of all possible gaits is quite high and it is not possible to check them all.

There are two phases during each leg movement, which are characteristic for legged robots: the swing phase and the stance phase (sometimes called return stroke and power stroke). There are also two remarkable positions of each leg [11, 13]: the anterior extreme position (AEP) – the position, when the leg is on the ground at the end of its swing phase (return stroke), and the posterior extreme position (PEP) – the position, when the leg is on the ground at the end of its stance phase (power stroke). In the swing phase the leg is moving from the PEP to the AEP and in the stance phase the leg is moving from the AEP to the PEP (see Figure 6).

The swing phase and stance phase are characterized with the length of their trajectories. The longer the trajectory of the swing and stance phases, the bigger the distance travelled by the leg on the ground. The different lenght of the swing and stance phases of opposite legs also leads to turning.

IV. Hexapod Controllers

When building a walking robot, one of the main parts is a controller – system, which determines the order and range of the leg movements. One approach is a exact mathematical model, which determines the movement and contact with the environment of each leg. This can be done using mathematical formulations or inverse kinematics models. However, this can be quite difficult, because completely modeling of all as-

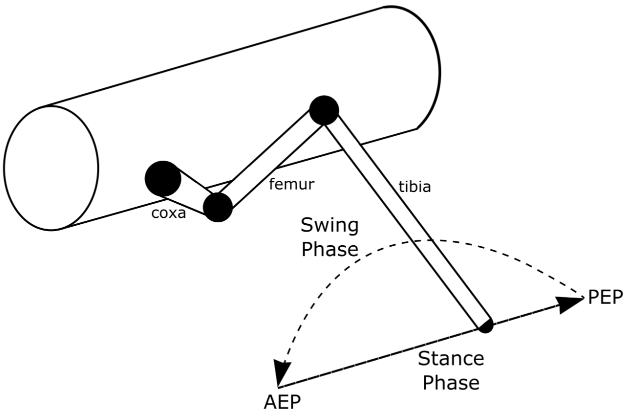


Figure. 6: Swing and stance phases of the leg. Each step of a leg consists of two phases: swing phase and stance phase. Swing phase is the movement of the leg to the the anterior extreme position (AEP) – the position, when the leg is on the ground at the end of its swing phase (return stroke) – and stance phase is the movement of the leg to the posterior extreme position (PEP) – the position, when the leg is on the ground at the end of its stance phase (power stroke). This figure is inspired by [11].

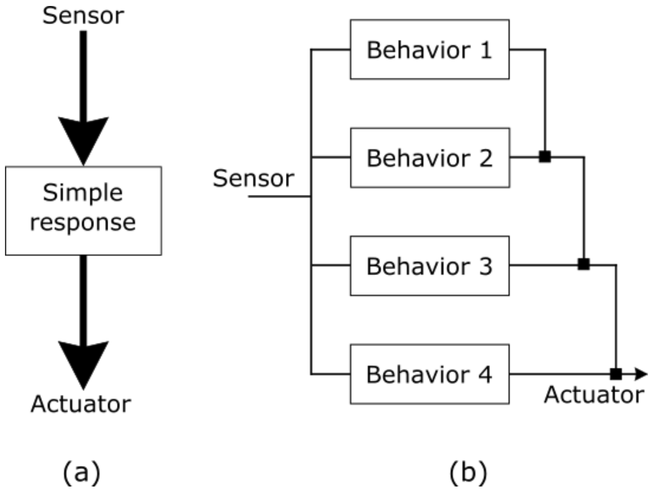


Figure. 7: (a) Reactive control architecture. The reaction of the robot to a specific sensor input is predefined. On a specific sense is assigned appropriate reaction. The advantage of the reactive architecture is its fast response to a change of the environment. The disadvantage is, that the architecture lack any planning. (b) Subsumption architecture. The architecture is based on priority behaviors organized into layers. The problem of this control architecture is the right order of the layers. This Figure is taken from [11].

pects of the robot and the environment and other influences is very complex task [11].
Therefore researchers are trying to develop some methods, which would generate the best gait according to given situation. Wide range of evolution techniques, such as neural networks or genetic algorithms, can be used to generate control pulses. Also biologically inspired approach central pattern generator (CPG) is based on neural networks, which can generate walking patterns.

A. Common Control Architectures

The robot control architectures are related to sensing, monitoring and acting actions of the robot. Different kinds of robot controllers can be distinguished [11]:

- Reactive and subsumption / behavior-based control architectures
- Deliberative controllers (hierarchical) or sense-plan-act control architectures
- Hybrid control architectures

The reactive control architecture (scheme based) is a stimulus-response based. The reaction of the robot to a specific sensor input is predefined – each action has a reaction. Although the reactive control architecture response speed is rather high, which can be an advantage when operating in real world where the response time is very important, the reactive architecture is not suitable for predictive planned outcomes. The reactive control architecture is shown in Figure 7a.
The subsumtion architecture (behavior based) is an alternative to the reactive system architecture. It is based on priority behaviors organized into layers. The lower layer behaviors (reflexes) can inhibit higher layer behaviors. The problem of this control architecture is the right order of the layers. The subsumption control architecture is in Figure 7b.
The deliberative control architectures are based on the Sense-Plan-Act principle and for their optimal functioning they usually need full knowledge about the environment. In the deliberative control architecture the robot first senses the environment, then creates a list of possible solutions. The robot also considers the results of the plans when choosing appropriate actions. The advantage of this architecture is that the goal can be easily achieved thanks to the goal oriented architecture. On the other hand this architecture is rather slow and it is not suitable for purposes where a quick reaction is needed. Also when the environment changes, the architecture must be changed too. This architecture is in Figure 8a.
The limitations seen by the reactive and the deliberative architectures can be solved by combining both approaches into a hybrid architecture. There are many different kinds of hybrid control architecture. In general, the hybrid architecture uses higher level planning in order to guide the lower level of reactive components. The advantage of hybrid architecture is the goal oriented architecture represented by deliberative layer and at the same time the reactive layer can execute low level actions. The hybrid control architecture is shown in Figure 8b.

B. Controllers Based on Neural Networks

One of possible approaches how design controllers is usage of neural networks. Beer [14] developed a recurrent neural network based on studies of the American Cockroach, but the neural network was tuned by hand to produce the desired results. Although results of his work are great success, it still has too much human interaction. Beer et al. proposed several papers on the field of walking robots. They created distribut-

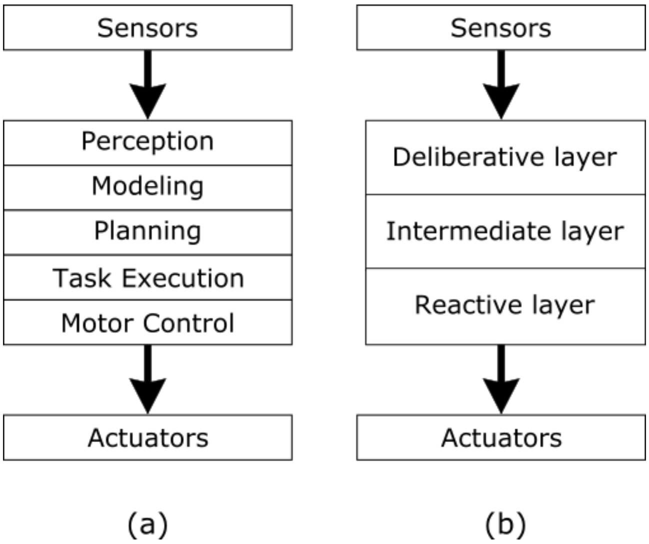


Figure. 8: (a) Deliberative control architecture. The architecture is based on Sense-Plan-Act principle and for its optimal functioning it usually need full knowledge about the environment. In the deliberative control architecture the robot first senses the environment, then creates a list of possible solutions. (b) Hybrid control architecture. Hybrid architecture is a combination of reactive and deliberative control architectures and it keeps the advandages of both architectures – planning and reactive approach. This Figure is taken from [11].

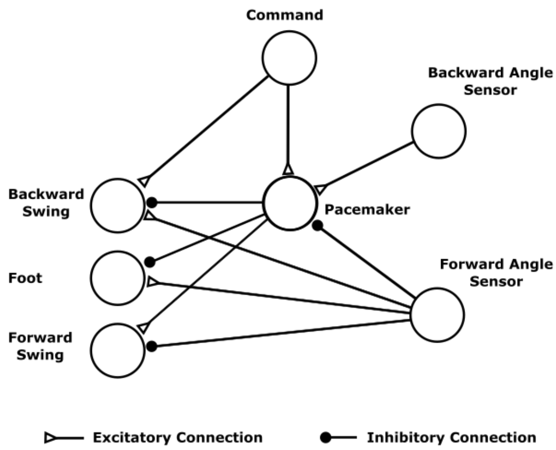


Figure. 9: The leg controller. Each leg is controlled by three motor neurons, which are driven by the pacemaker neuron whose output rhythmically oscillates. A single command neuron makes the same two connections on every leg controller. The forward angle sensor can inhibit the pacemaker neuron and the backward angle sensor can excite pacemaker neuron and change its rhythm. This figure is taken from [15].

ed neural network based on insects neurology. In [15] they have presented a fully distributed neural network architecture designed for hexapod robot control. The design of the network is based on work on the neuroethology of insects locomotion. The controller was tested in simulation in previous work. They report in this paper, that they successfully applied the controller on a real hexapod robot. The results were quite similar to the results observed in the simulation. The robot is capable of movement using different gaits. The controller is shown in the Figure 9. Each leg had its own controller, which operates in following manner: Normally, the foot motor neuron is active (supporting the robot body). When the command neuron excites the backward swing motor neuron, the leg is moved backward (stance phase). Periodically, the pacemaker neuron interrupts the stance phase and excites the forward swing motor neuron (swing phase). The frequency of pacemaker bursts and the velocity output of the backward swing motor neuron depend on the level of excitation provided by command neuron. Additionally, sensors can reset the pacemaker neuron. Adjacent pacemakers mutualy inhibit one another to ensure that adjacent legs will not swing at the same time.

Chiel et al. [16] discuss the robustness of the controller based on the used gait. The robot was capable of stable movement at slow, medium and fast gaits with disconnected forward or backward angle sensor of any leg. Also removing the connections between pacemaker neurons did not prevent the robot from walking stably at any speed. Finally, after disabling the lift motor of the middle leg and retracting the leg so it does not supported any load, the robot was capable of stable walk at the slower gaits, but the robot was unable to walk using the fastest gait, because the tripod gait requires the middle leg. If the leg was disabled, but it was allowed to contact with the ground, the robot turned toward the side with the disabled leg.

Suitable inspiration for the design of the controller is the movement of animals. Beer et al. [17] 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 insects approach. Espenschied and Quinn [18] 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.

Goldschmidt et al. [19] present an adaptive neural control mechanism allowing hexapod robots to negotiate obstacles. The solution was tested in simulated environment and on a real robot (AMOS II) and the results of testing shows that the robot can efficiently negotiate obstacles with a height up to 85 % of the robot leg length in simulation and 75 % in a real environment.

Studies of animal nervous system show that the pattern of locomotion is controlled by neural centers located in the neural systems below the brain stem in the spinal cord known as central pattern generators (CPGs), whose output is an oscillating signal with a certain frequency. The concept of CPG came from experiments, which demonstrated that a group of neurons could produce a rhytmic pattern while isolated from any sensor input [20]. Mathematical models for CPG are proposed in [21], where the CPGs consisted of two neurons. These CPGs are also widely used to generate control signals for walking chassis. Ijspeert et al. [22] present a spinal cord model. They address three fundamental issues related to ver-

tebrate 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.
Yu et al. [23] propose a novel CPG-based control architecture for hexapod walking robot. They divided the motion control into the gaits generation level and joints coordination level. The first level is implemented using CPG network in ring based on modified Van der Pol oscillator. The second level they address the problem of multi-DoF coordination of a single leg through phase order modulation and amplitude adjustment of the neural oscillators. Each leg has its own controller, which provides rhythmic signal. Six of these controllers are connected and produce periodic signals with identical amplitude and frequency, but the phase difference of each controller is precisely shifted, so they produce desired gait. The gait transition can be understood as the controller has the ability to recover from the initial condition "out of phase". The authors also present the results of testing the controller on a real robot.
Barron-Zambrano et al. [24] present a CPG-based controller for quadruped and hexapod walking robots, which can generate several gaits. The proposed implementation of the controller is modular and configurable so it can control legged robots with different number of degrees of freedom. The controller is implemented on an embedded Field Programmable Gate Array (FPGA). A method based on genetic algorithm was used to find the parameters of CPG.
Chung et al. [25] proposed a CPG-based control strategy for hexapod walking robot. The CPG controller uses the Matsuoka's neural oscillators. Each oscillator consists of two neurons mutually inhibiting each other. The controller uses an inertial measurement unit to get the attitude of the body and to generate the control signals accordingly. The controller was successfully tested on a real robot in irregular terrain.
The problem with CPGs is connecting the data from sensors to it. It is possible, however, it is quite complicated, because the signal from the sensor may come at any state of the step and the controller may not be able to handle it. The solution is to wire the output of CPGs to another neural network, which controls and modulates the output of CPGs based on the data from the sensors. Barron-Zambrano et al. [26] modulate the output of CPG using fuzzy logic approach, which manages gait speed modulation and direction control, and finite state machine, which selects gait and manages transitions among them.
Parker and Lee [27] suggest to learn individual legs separately and then connect the individual neural networks together. At first, a small network is formed, which is able to control the movement of one leg. This network has no connection to the other legs and is able to generate pulses independently of the other legs. These individual networks are then connected in one large network, which controls whole gait generation.

C. Neural Networks Training Approaches

Standard methods, such as backpropagation, can be used to find appropriate weights of individual neurons in the network. But there are also other possibilities. Parker and Lee [27] used genetic algorithms for learning neural networks. The network structure is created (individual neurons and their connections), and then genetic algorithms are used to create descendants, who represent the weight vectors of the individual neurons.
Neural network prepared by this way is started and it generates hundreds of control pulses for the motors. The generated pulses are then evaluated by the fitness function, which has three basic parameters. The first parameter is the forward motion, which corresponds to the movement when the leg is placed on the ground. The second parameter is the number of leg lifts. It is a penalty, because lifting leg does not move forward and needlessly consumes energy. The third parameter is the resistance, which is a penalty which occurs when a leg is set in the rearmost position and is placed on the ground. Such limb merely slows forward movement.
Also other researchers use genetic algorithms. Lewis, Fagg and Bekey in [28] describe staged evolution of a central pattern generator (CPG) for movements control of a hexapod robot. The CPG is designed as a neural network. But insted of using a learning algorithm to train the neural network they used genetic algorithm to alter the interconnection weights of the neural network. The same controller is described in [29]. Parker and Rawlins [30] introduced cyclic genetic algorithms, which can be used to gait generation.

D. Inspiration in the Nature

Almost all approaches used in the design and control of walking robots are inspired in 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 inspired by nature, we can study how animals solve difficult situation during their movement.
The application of biological methods and systems found in nature to the study and design of engineering systems and modern technology is called Bionics [11]. Besides the term Bionics (from biology and electronics) it can be also used Biomimetics or Biomimicry (from bios = life, and mimesis = to imitate). Bionics is related to applying ideas seen in nature for solving technical, engineering or scientific problems. Bionics is rather inspired by the solution observed in nature instead of mimic the structure behind it. This is more close to Biomimetics.
The design of a legged robot like hexapod shows a practical use of Bionics for the robotics. Depending on the number of legs the design of a walking platform can be inspired by insects in the case of six-legged robots, spiders in the case of eight-legged robots or animals like horse in the case of four-legged robots.

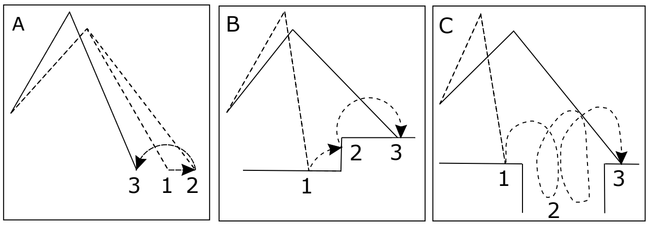


Figure. 10: A) Stepping reflex. The leg can step from the position (2) to the position (3) to better support the body. B) Elevator reflex. If the leg encounters an obstacle (2), it tries to lift the leg higher to step over the obstacle. C) Searching behaviour. If the leg cannot reach ground at expected location (2), it tries to find another foothold (3). This figure is taken from [34].

Zhang et al. [31] developed a bionic hexapod, which is capable of walking on unstructured terrain. The controller uses the Posture Control strategy based on Force Distribution and Compensation (PCFDC) [32]. The experimental results show that the robot can keep stable while walking over unstructured terrain.
Chen et al. [33] analyzes the movement of insects and they use the results in the construction of a walking robot. The results show that the proposed leg structure is able to perform effective swing movements on rough terrains.
Espenschied, Quinn, Beer and Chiel [34] proposed using reflexes, which were observed in insects. When the leg moves forward, stepping, elevator and searching reflexes are used to find suitable position for the leg. The reflexes are shown in Figure 10.
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.
Ferrell [35] 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.

E. Other controllers

Besides of bio-inspired controllers, we can use also classic controllers which have predefined movements for each leg. These controllers handle unknown situations much harder, because they cannot change themself and adapt to the new situations.
Ali et al. [36] use a FPGA for generating control pulses for a hexapod robot. The FPGA drives continuous servomotors using pulse width modulation (PWM). Using a FPGA decrease the power consumption of the controller and the required chip space.

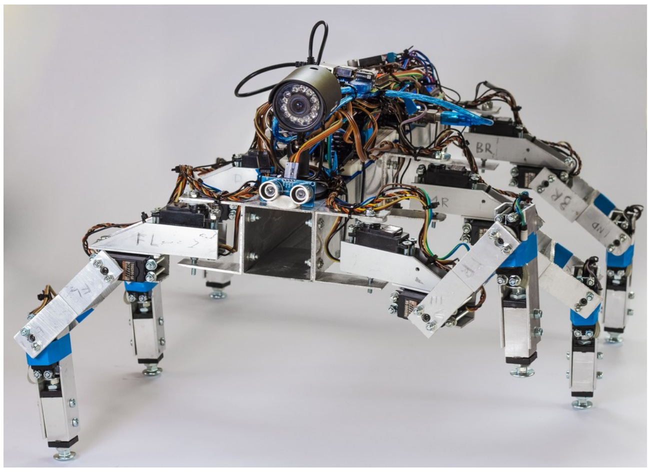


Figure. 11: Hexapod robot of our design. This robot was designed and constructed during the project. It is equipped with sonars, camera, LCD display, force-sensitive resistors, encoders and more accessories. The robot is capable of movement using many gaits including tripod, wave and ripple. The robot is constructed of aluminium profiles, is powered by one Li-Po accumulator and can operate in rough terrain. Each leg is equipped with three degrees of freedom. Servomotors HS-5485HB and HS-5645MG are used to move robot legs.

V. Our Robot

In our future work we want to continue in the research of controlling hexapod robot using evolution techniques. Therefore we build our hexapod robot so we can test our solutions (Figure 11). Our robot is build of aluminum profiles and has 18 hobby servomotors (each leg has 3 degrees of freedom). The servomotors are equipped with encoders and each leg has a ground sensor, which can detect obstacles or ground under the leg during step. To detect the ground we use force-sensitive resistors, which are better than tactile sensors, because the value from force-sensitive resistor can be used to distribute the weight of the robot to all legs equally.
The robot is capable of movement in rough terrain and can use common gaits. Unlike commercial versions of walking robots, which can be purchased, this robot has more sensors and can be easily extended. More information about our robot and several videos can be found at http://hexapod.marekzak.cz.

A. Robot Electronic System

The robot is controlled by microcontroller Atmega2560[1], which is integrated on the Arduino Mega 2560 platform, and Raspberry Pi board[2] – a miniature computer the size of a credit card. It has extremely low power consumption (max. 3.5 W) and can run linux based operating system Raspbian. We use Raspberry Pi model B+, which provides enough computing power to run more complicated calculations. The Raspberry Pi is equipped with a camera and USB Wi-Fi dongle, which provides connection to a wireless network. The

[1] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf, visited 20-01-2016.

[2] http://www.raspberrypi.org/, visited 20-01-2016.

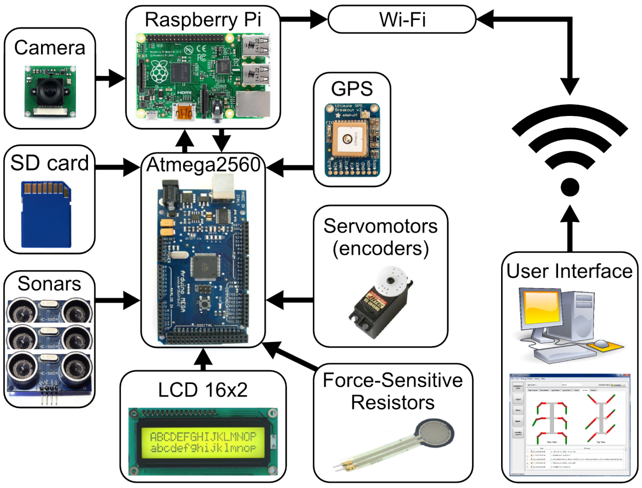


Figure. 12: The electronic system of our robot. In the center is a MCU Atmega2560 integrated on an Arduino Mega 2560. Most of the sensors like sonars, LCD display, memory card, GPS module or force-sensitive resistors are connected to it. There are also 18 servomotors connected to digital pins and driven by MCU's 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.

robot is also equipped with ultrasonic sonars to detect obstacles, LCD display, which displays basic information about the robot, SD card and GPS module for outdoor navigation. The scheme of the electronic system is in Figure 12.

The whole system is powered by one (or optionaly two) 11.1 V Li-Po accumulator, which can supply 60 A, which is enough for the servomotors and electronic system. The robot is equipped with voltage regulators to transform the voltage to 5 V for the electronic and 6 V for the servomotors. We use switching voltage regulators to decrease power consumption. The robot can operate approximately two hours on one accumulator, based on the selected gait and other conditions.

B. *Robot Control Software*

The robot control software consists of three parts: Hexapod Control Room, which is UI program to control and monitor the robot, Robot Client, which is program for Raspberry Pi and program for the microcontroller. The programs are writen in C++ and the UI uses Qt library[3].

The Hexapod Control Room is an user interface program (UI), which allows to control the robot and visualizes actual positions of the legs and data from sensors. Up to ten robots can be connected to the UI and user can switch among them to see data from sensors and to control the selected robot. The UI has a tool to generate custom gaits, which can be simulated within the UI or directly on the robot. The UI includes a console which can be used to send commands directly to the robot MCU (see Figure 13).

The Robot Client program is high level controller of the robot. It receives messages from the computer over the Wi-Fi

[3]http://doc.qt.io/

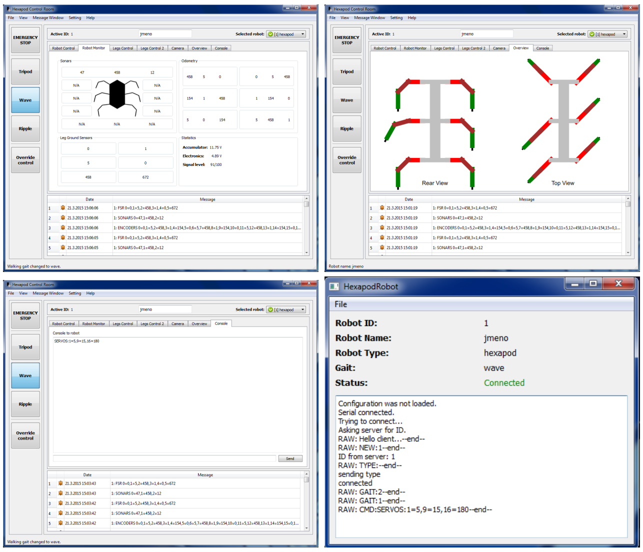


Figure. 13: User interface screenshots of Hexapod Control Room. Top left figure shows the Monitor screen with the values of sonars, encoders and force-sensitive resistors. There is also information about battery level and robot wi-fi signal strength. Top right image shows the actual position of the legs. 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. A logging window is on the bottom of the UI, which displays messages from the robot and ather information about the robot and UI state. Bottom right image shows the client window with the most important information and log window.

and commands the microcontroller to perform them. It also receives data from the microcontroller and sends them to the control computer.

The last but not least part of the software is the program for the microcontroller, which controls the low level systems like servomotors, ultrasonic sonars, LCD display, force-sensitive resistors and other peripherals, and communicates over a serial line with the Raspberry Pi. The steps of each gait are stored in a calendar structure and execute in specific time, because the control of servomotors is quite demanding, but the main loop must stay non-blocking. The algorithm of the step execution is similar to the next-event algorithm.

VI. Conclusions

This paper deals with overview of several controllers for hexapod robots designed using evolution techniques, such as neural networks or genetic algorithms. Researches show, that central pattern generators are very suitable to generate control signals for legged robots. We discuss several characteristics of legged robot, their most common gaits and we describe possible ways of inspiration in nature when designing or controlling a legged robot. We also mention several existing walking robots and we introduce a hexapod robot of our design. Currently we are working on a simulation environment. An exact 3D model (Figure 14) 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 hexapod controller.

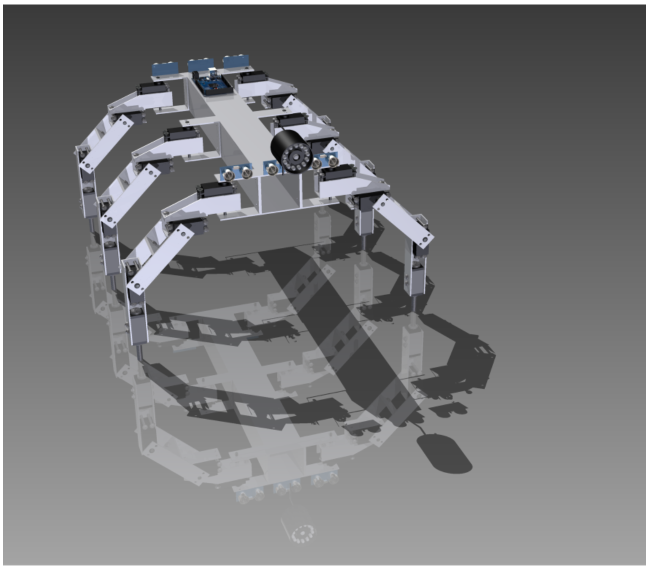


Figure. 14: The 3D model of the robot.

Acknowledgments

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] U. Saranlı, "Dynamic locomotion with a hexapod robot," Ph.D. dissertation, The University of Michigan, 2002.

[2] G. Bekey, *Autonomous robots : from biological inspiration to implementation and control*. Cambridge, Mass: MIT Press, 2005.

[3] F. Delcomyn, "Walking in the american cockroach: the timing of motor activity in the legs during straight walking," *Biological cybernetics*, vol. 60, no. 5, pp. 373–384, 1989.

[4] NASA, "All-terrain hex-limbed extra-terrestrial explorer," http://athlete.jpl.nasa.gov/, 2016, [Online; visited 20-01-2016].

[5] Boston Dynamics, "Boston dynamics: Dedicated to the science and art of how things move." http://www.bostondynamics.com/robot_ls3.html, 2016, [Online; visited 20-01-2016].

[6] ——, "Boston dynamics: Dedicated to the science and art of how things move." http://www.bostondynamics.com/robot_rhex.html, 2016, [Online; visited 20-01-2016].

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

[8] E. Moore and M. Buehler, "Stable stair climbing in a simple hexapod robot," DTIC Document, Tech. Rep., 2001.

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

[10] 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.

[11] B. Jakimovski, *Biologically inspired approaches for locomotion, anomaly detection and reconfiguration for walking robots*. Berlin: Springer, 2011.

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

[13] M. Schilling, T. Hoinville, J. Schmitz, and H. Cruse, "Walknet, a bio-inspired controller for hexapod walking," *Biological cybernetics*, vol. 107, no. 4, pp. 397–419, 2013.

[14] R. D. Beer, *Intelligence As Adaptive Behavior: An Experiment in Computational Neuroethology*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[15] R. D. Beer, H. J. Chiel, R. D. Quinn, K. S. Espenschied, and P. Larsson, "A distributed neural network architecture for hexapod robot locomotion," *Neural Computation*, vol. 4, no. 3, pp. 356–365, 1992.

[16] H. J. Chiel, R. D. Beer, R. D. Quinn, and K. S. Espenschied, "Robustness of a distributed neural network controller for locomotion in a hexapod robot," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 293–303, 1992.

[17] 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.

[18] 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.

[19] D. Goldschmidt, F. Wörgötter, and P. Manoonpong, "Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots," *Frontiers in neurorobotics*, vol. 8, 2014.

[20] A. I. Selverston, "Are central pattern generators understandable?" *Behavioral and Brain Sciences*, vol. 3, no. 04, pp. 535–540, 1980.

[21] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological cybernetics*, vol. 56, no. 5-6, pp. 345–353, 1987.

[22] 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.

[23] H. Yu, W. Guo, J. Deng, M. Li, and H. Cai, "A cpg-based locomotion control architecture for hexapod robot," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5615–5621.

[24] J. H. Barron-Zambrano, C. Torres-Huitzil, and B. Girau, "Configurable embedded cpg-based control for robot locomotion," *International Journal of Advanced Robotic Systems*, vol. 9, 2012.

[25] H.-Y. Chung, C.-C. Hou, and S.-Y. Hsu, "A cpg-inspired controller for a hexapod robot with adaptive walking," in *Automatic Control Conference (CACS), 2014 CACS International*, Nov 2014, pp. 117–121.

[26] J. H. Barron-Zambrano, C. Torres-Huitzil, and B. Girau, "Perception-driven adaptive cpg-based locomotion for hexapod robots," *Neurocomputing*, p. 17, 2015.

[27] G. B. Parker and Z. Lee, "Evolving neural networks for hexapod leg controllers," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1376–1381.

[28] M. A. Lewis, A. H. Fagg, and G. Bekey, "Genetic algorithms for gait synthesis in a hexapod robot," *Recent trends in mobile robots*, pp. 317–331, 1994.

[29] M. A. Lewis, A. H. Fagg, and A. Solidum, "Genetic programming approach to the construction of a neural network for control of a walking robot," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 2618–2623.

[30] G. B. Parker and G. J. E. Rawlins, "Cyclic genetic algorithms for the locomotion of hexapod robots," in *Proceedings of the World Automation Congress (WAC96)*, vol. 3, 1996, pp. 617–622.

[31] H. Zhang, Y. Liu, J. Zhao, J. Chen, and J. Yan, "Development of a bionic hexapod robot for walking on unstructured terrain," *Journal of Bionic Engineering*, vol. 11, no. 2, pp. 176–187, 2014.

[32] H.-q. LIU, Y.-b. LIU, H. ZHANG, J. CHEN, and J. ZHAO, "Posture control strategy based on force distribution for bionic six-legged robot [j]," *Journal of Harbin University of Commerce (Natural Sciences Edition)*, vol. 5, p. 009, 2012.

[33] J. Chen, Y. Liu, J. Zhao, H. Zhang, and H. Jin, "Biomimetic design and optimal swing of a hexapod robot leg," *Journal of Bionic Engineering*, vol. 11, no. 1, pp. 26–35, 2014.

[34] 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.

[35] 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.

[36] N. A. Ali, S. I. M. Salim, R. A. Rahim, S. A. Anas, Z. M. Noh, and S. I. Samsudin, "Pwm controller design of a hexapod robot using fpga," in *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*. IEEE, 2013, pp. 310–314.

Author Biographies

Marek Žák was born in Brno on 7. 10. 1989. He entered the Faculty of Information technology of the Brno University of Technology in 2009 where he studied intelligent systems and robotics. He received M.Sc. in 2015. Currently he is studying Ph.D. at the Faculty of Information Technology of Brno University of Technology; his thesis title is "Bio-inspired controllers for hexapod robots". His main research interests are robotics, artificial intelligence and bio-inspired systems.

Jaroslav Rozman was born in Brno on 3. 11. 1979. He entered the Faculty of Electrical Engineering and Computer Science of the Brno University of Technology in 1998 where he studied cybernetics and automation. He received M.Sc. in 2003. He defended his Ph.D. in the field of robotics in 2011 at the Faculty of Information Technology of Brno University of Technology; his thesis title was "Mobile Robot Navigation". Currently he is working as Assistant Professor at the Department of Intelligent Systems. His main research interests are robotics, artificial intelligence and computer vision.

František V. Zbořil was born on 10. 11. 1945. He is an associate professor of Computer Science at the Faculty of Information Technology, Brno University of Technology, Czech Republic. He received his Ph.D. in 1978 at the same university. He started his research activities on analogue and hybrid computers with simulation of continuous systems. His next research was focused on classical artificial intelligence, robotics and neural networks. Now, the main objects of his professional interests are soft computing problems. He is the author of more than 100 papers and he is a member of several educational, research and academic boards or societies.