Jan Czarnowski 226869
Jan Fraś 226890

30.01.2013

Designing, building and programming a hexapod robot

The basic problem of mobile robotics is locomotion. Robots can achieve it in many different ways – some are based on nature, other invented by humans. Choosing one method usually defines the robot's degree of usability. The authors' aim is to develop a machine which can effectively move through various types of challenging terrain. This paper presents the design and implementation of a machine which tries to fulfill that goal – a hexapod, spider-like robot. The paper is split into 10 chapters; each of them covers one aspect of robot's hardware or software. Starting with basic concepts, each chapter introduces a higher level aspect – construction, algorithm etc., ending with high level control interface presented to the user. The first chapter contains a brief introduction to the problem – why effective locomotion is important in mobile robotics and what is the aim of the work.

In the second chapter, analysis of various methods of locomotion has been presented. It analyses advantages and disadvantages of different animal gaits, wheels and continuous tracks. Despite being the most energetically efficient, the last two solutions are not sufficient for moving in difficult terrain. For this reason, walking has been chosen in order to be used in our robot. There are two types of walk – dynamically and statically stable. The differences between them are presented and examined for suitability to use in the project. The chapter also explaines the choice of statically stable, six legged walk.

After choosing proper locomotion method, proposed construction is introduced in the next chapter. A kinematic chain of hexapod's leg design is also presented. It allows reaching any point in sphere constrained by the parts' length and achieving any trajectory within. This, in turn, allows the robot to move through uneven terrain and go over obstacles. The leg's kinematic model is analysed and equations connecting joint angles with the position of leg tip in coordinate system with origin on robot's chasis are derived. The chapter also contains calculation and graphs of forces and moments acting on robot parts when it is moving.

The fourth chapter describes movement algorithms. The main parameters taken into consideration are speed and fluency of motion. Maintaining stability of robot's chasis is also an important aspect, as the robot will likely carry a camera on board. The most optimal sequence of walk is chosen. Respective equations are derived. Combined with transformations from the previous chapter, they are required for generating control signals for motors. These informations were used to develop program for controlling the robot on the lowest level: ARM microcontroller connected with servomechanism driver. It is presented in the fifth chapter, which also describes the used hardware.

The next chapter describes the robot's control system and devices that it consists of. Implementing high level functionalities like camera transmission, voice recognition or image processing based autonomy need more computational power and resources. Because of that, a third device is introduced – PandaBoard - a system on chip running High Level Operating System (HLOS). This chapter focuses mainly on brief description of used devices and comunnication channels between them.

Chapter seven presents the PandaBoard more closely. It describes the choice of HLOS for the device and its role. The process of OS installation on Embedded Systems is also presented. After installation, additional actions were required in order to configure the device. This tasks included

installing kernel patch to enable chip's features, enabling built-in onboard hardware decoder/encoder system and configuration of X windows. Nessesary software used for camera transmission over the RTP protocol is also presented.
The next chapter introduces the Robot Operating System (ROS) software used on the PandaBoard and client computers. It explains the need of such software and its role. Based on information from creators' website, the system functioning is briefly described. The chapter is concluded with information on compilation ROS for ARM architecture.

Chapter nine presents programs created for robot control. Three modes are available: idle mode, joystick control and autonomy mode. A state machine has been implemented for managing robot states, using mechanisms provided by ROS. The autonomical state causes the robot to follow a special marker. To achieve this, computer vision algorithms had to be programmed. This chapter explains in detail how these programs work.

The paper ends with conclusions drawn from building the robot and final thougths. This part sums up all the capabilities of our hexapod.