CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's thesis

Online Foot Strike Detection of a Hexapod Walking Robot Using Inertial Measurements

Jiří Kubík

May 2018

Supervisor: Doc. Ing. Jan Faigl, Ph.D.
Supervisor specialist: Ing. Petr Čížek

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kubík Jiří** Personal ID number: **457189**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

Branch of study: **Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Online Foot Strike Detection of a Hexapod Walking Robot Using Inertial Measurements

Bachelor's thesis title in Czech:

Detekce došlapu šestinohého kráčejícího robotu

Guidelines:

1. Familiarize yourself with the hexapod walking robot platform of the Computational Robotics Laboratory and its locomotion control [1].
2. Familiarize yourself with methods of processing and classification of inertial measurements for foot strike detection[2-5].
3. Select and implement a suitable method for foot-strike detection from inertial data.
4. Compare the developed method with the foot-strike detection approach based on force feedback of servo drives [1].
5. Investigate the possibility of using the inertial measurements for terrain classification.

Bibliography / sources:

[1] J. Mrva, J. Faigl, "Tactile sensing with servo drives feedback only for blind hexapod walking robot", in 10th International Workshop on Robot Motion and Control (RoMoCo), 2015, pp. 240-245.
[2] A. T. M. Willemsen, F. Bloemhof and H. B. K. Boom, "Automatic stance-swing phase detection from accelerometer data for peroneal nerve stimulation," in IEEE Transactions on Biomedical Engineering, vol. 37, no. 12, pp. 1201-1208, 1990.
[3] F. L. Garcia Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel and R. S. Fearing, "Performance analysis and terrain classification for a legged robot over rough terrain," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 513-519.
[4]] K. Walas, D. Kanoulas and P. Kryczka, "Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing," in IEEE-RAS 16th International Conference on Humanoid Robots, 2016, pp. 133-140.
[5] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 5603-5608.

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Jan Faigl, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **12.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

doc. Ing. Jan Faigl, Ph.D.	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Ing. Pavel Ripka, CSc.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Declaration

I declare that the presented work was developed independently ant that I have listed all sources of the information used within it in accordance with the methodical instructions for observing the ethiclal principles in the preparation of university theses.

Prague, May 25, 2018

. . . . . .
Jiří Kubík

Acknowledgement

I would like to thank my supervisor doc. Ing. Jan Faigl, Ph.D. and my supervisor specialist Ing. Petr Čížek for their guidance, advises and patience with me. But most of all, I would like to thank them for the knowledge and experience they gave me. Besides, I would like to thank my loved ones for their support during my whole studies because it would not be possible without them.

Abstrakt

V této práci zkoumáme možná využití inerciálních měření k detekci došlapů šestinohých kráčejících robotů. S využitím relativně levných akcelerometrů je možné vytvořit detektor došlapů, který ummožňuje šestinohému robotovi překonávat i nerovné terény. Kromě detekce došlapů také zkoumáme možná využití akcelerometrů ke klasifikaci různých druhů terénů. Z měřících módů, které poskytují použité akcelerometry, jsme vybrali dva: standardní plynulý měřící mód a mód založený na přerušeních, který naměřená data filtruje a vybíra ta, která splňují určité požadavky. Detektory došlapů založené na předfiltrovaných datech dosahují lepších výsledků, než detektory využívající kontinuálně měřená data. Míra abstrakce, které detektor došlapů dosáhl, mu umožňuje detekovat došlapy i při zvýšení celkové rychlosti pohybu navzdory tomu, že detektor byl naučen na mírně odlišných akcelerometrických datech nasbíraných při nižších rychlostech. V porovnání s předchozí metodou detekce došlapů inspirovanou měřením torzních sil v kloubech robota se jeho celková rychlost zvýšila jak v případě rovného terénu, tak v případě terénu nerovného. Detektor došlapů i terénní klasifikátor jsou testovány na šestinohém kráčejícím robotovi.

Klíčová slova: Kráčející roboti, Detekce došlapů, Inerciální měření

Abstract

In this thesis, the possible usage of the inertial measurements for the foot-strike detection of the hexapod walking robot in rough terrains is examined. We show that affordable accelerometers attached to robot's legs can be utilized for a reliable foot-strike detection and thus allow the robot to crawl irregular terrains. Furthermore, we examine the possible advantages of two operation modes of the accelerometers and the underlying foot-strike detection pipeline. Namely, the foot-strike event detection in a continuous stream of data from accelerometers. And interrupt-based mode, in which the accelerometers filters data at the hardware level to provide only relevant data around the foot-strike event. We also investigate the possibility of using the accelerometers for the terrain classification. The proposed utilization of the interrupt-based mode provides significantly better performance. The presented foot-strike detector supports generalization from the data collected using the slow locomotion to faster locomotion where the inertial signals slightly change. The speed of the hexapod walking robot in both flat and rough terrains is improved in comparison to the adaptive motion gate that uses a force threshold-inspired position controller for the foot-strike detection. Both the foot-strike detection and the terrain classification have been experimentally tested using a real hexapod walking robot.

Keywords: Legged robots; Foot-strike detection; Accelerometric measurements

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Robotics is nowadays rapidly evolving research field with many practical deployments. Advancements in miniaturizations and energy efficiency help to push mobile robotic research forward. The primary objective of mobile robotics is to develop intelligent and autonomous systems and deploy them in increasingly complex scenarios and environments. Mobile robots can be divided according to locomotion mechanism into three main categories; wheeled, tracked, and legged robots. The main advantage of legged robots over wheeled or tracked robots is a capability of traversing rough terrains in a non-destructible way (terrain negotiation), for the price of more complex locomotion control methods that control robot legs when crawling over rough terrains.

Different locomotion control methods for walking robots have been presented in recent years, but they all share a common property of utilizing feedback control from sensors to negotiate irregular terrain and to maintain the robot stability. Even though mechanical construction of the legged robots providing more degrees of freedom may improve its ability to traverse rough terrain [1], locomotion control system is crucial to achieving smooth and non-hazardous locomotion. Hence studying and progressive improvement of locomotion control is crucial for improving robot behavior and its performance in various scenarios, accomplishing the desired mission objectives as well as increasing the overall locomotion speed. A speed of the locomotion is one of the key factors for the robot performance; fast locomotion control allows a walking robot to accomplish given tasks more quickly and therefore improve its overall performance in the missions. Thus the primary motivation for this thesis is to provide fast and reliable locomotion over rough terrain for affordable hexapod walking robot using as few newly added sensors as possible.

During the locomotion, a leg either move to new foothold (swing phase) or support the body (stance phase). Several problems may arise after missed foot-strike (transition between swing phase and stance phase). Firstly, missed foot-strike may results in a situation, when most of the robot's mass is supported by fewer legs than expected, which put under stress both actuators and construction. Stressed actuators require more electric current to flow through them, which may result in an overheating and may lead to failure of servo and lost their ability to support the body. Secondly, after the missed foot-strike, the body is asymmetrically raised, which may move the center of mass out of expected area and possibly result in stability loss. The loss of the stability may cause robot falling and damage. Therefore reliable and timely detection of foot-strike is critical for both maintaining the stability of the robot and preventing the robot from being damaged.

Our work is based on robotic platform presented by J. Mrva and J. Faigl in [2], where the authors have employed easy-to-use hexapod walking robot platform using feed-back solely from 18 Dynamixel AX-12A servos. Even though this suggested approach provides a reliable and relatively affordable solution for the foot-strike detection, it is, unfortunately, limited by the servo communication speed resulting in relatively slow speeds of locomotion. Therefore, in this thesis, we suggest extending the existing hexapod platform by adding inertial sensors used for the foot-strike detection enabled by specific features of the used accelerometers. We also aim to reach the same foot-strike detection reliability as the groundwork and if possible, speed up the overall locomotion of the hexapod. The original intentions of the groundwork are to provide a relatively cheap solution, hence to stick with these ideas we suggest to use six affordable 3-axial ADXL345 accelerometers attached to an individual leg of the robot along with a central reference heading system mounted on the robot trunk. The used accelerometers provide eleven operation modes and features. We decided to exploit

properties of the continuous *stream mode* and event-based *single-tap mode*. The stream mode continuously provides data, whether the single-tap mode filters data at the hardware level and provides only relevant data around the particularly configured event.

Our approach suggests using motion controller developed by the authors of [2], further referred as the *adaptive gait*, to supervise learning of signal detectors for foot-strike detection based on inertial feedback. Even though the adaptive gait is relatively slow, it was already proven it is capable of traversing irregular terrains. In presented results, we show, that our best performing foot-strike detector is sufficient enough to allow a robot to crawl over rough terrains. Furthermore the proposed signal classifier learned on the relatively slow adaptive gait is capable of keeping its properties even if the overall speed of gait cycle is increased, demonstrating its generalization capabilities. The suggested approach provides up to 1.7 times speedup of the groundwork approach [2] in the irregular terrain and up to 1.5 times speedup of the groundwork approach in the flat terrain. Aside from the foot-strike detection, the possible usage of the inertial measurements for the terrain classification has been examined.

The remainder of the thesis is organized as follows. The problem addressed together with the description of the hexapod locomotion, differences between the gaits and groundwork [2] are provided in the following section. The related approaches are outlined in Chapter 3 along with the description of the used classification methods. The hexapod platform and the employed sensors are presented in Chapter 4 with the emphasis on the used accelerometers. The proposed approach using inertial measurement for the terrain detection is described in Chapter 6. Results of two validation scenarios; the proposed classification methods success rate and the comparison to the previous work [2] are reported in Chapter 6 followed by the conclusion in Chapter 7.

Chapter 2
Problem Statement

Multi-legged robots are usually more complex than wheeled or tracked robots. For example, our platform has 18 controllable degrees of freedom (DoF) that is far more than controlling steering and and the forward velocity in a regular car. The complexity of the multi-legged enhances capabilities of the robot terrain negotiation, but the robot also requires more advanced control. One possible way to control multi-legged robots is to use defined motion pattern - gait. Gait control distinguishes between two phases of locomotion for each leg. A leg in the *swing phase* moves to a new foothold, and on the contrary, a leg supports the robot body in the *stance phase*. [3] During the robot movement, the robot legs alternate between the swing phase and stance phase.

Leg movement patterns can be distinguished by the number of legs supporting body (implying the number of moving legs) in each step of a gait cycle. According to the number of simultaneously moving legs, n-pod gaits are defined. Different n-pod gaits vary in the overall speed of the hexapod and its stability. At least three legs are required to support the body to keep the robot stable. This restriction allows only three possible n-pod gaits; *tripod* ($n = 3$), *tetrapod* ($n = 4$) and *pentapod* ($n = 5$).

The ***tripod*** gait splits legs into two triplets. One triplet alternates another while three legs are moving and three legs are supporting the body. The most stable triplets consist of the front and the rear leg from one side of the hexapod and the middle leg from the other side. Each gait cycle consists of only two steps, which makes the tripod gait fastest gait from the all mentioned above at the cost of losing stability in cases of failure of any supporting legs.

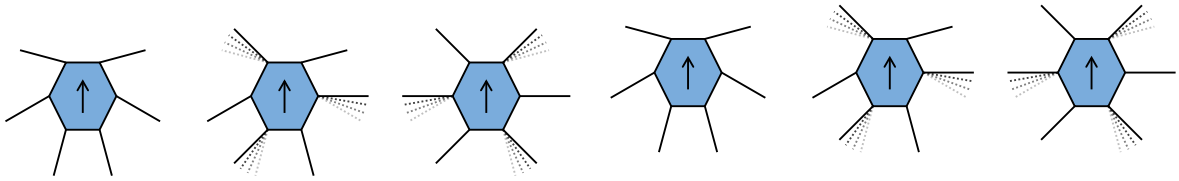


Figure 1: The visualization of the six steps of the tripod gait. In step 1., all legs are in the old footholds. Three legs are moving to the new footholds in step 2., and the other triplet is moving to the new positions in step 3. The steps 4., 5., and 6. are similar to the steps 1., 2., and 3.

The ***tetrapod*** gait defines three pairs of legs. Two pairs (four legs) support body, while one pair is moving. There are several combinations of legs in pairs and the order of the alternation. The most stable combinations require each leg from different sides of the hexapod and variation in the front-rear position. An example of one possible combination is (left-front,right-middle), (left-middle,right-rear), (left-rear, right-front). The tetrapod gait cycle consists of three individual steps of each pair of legs, which makes it slower than the tripod gait, but it increases stability by adding a leg to support body.

The ***pentapod*** gait operates with each of the hexapod legs individually, meaning only one leg is moving, while five remaining legs support the body. The order of legs in the pentapod gait does not affect the overall stability of hexapod, but most common approach uses the clockwise or counter-clockwise order of the leg alternation in the gait cycle. The pentapod gait requires six steps in each one of it, and therefore it is the slowest, but the most stable one.

All n-pod gaits are periodical gaits, and the legs are moved in the same order. There are also

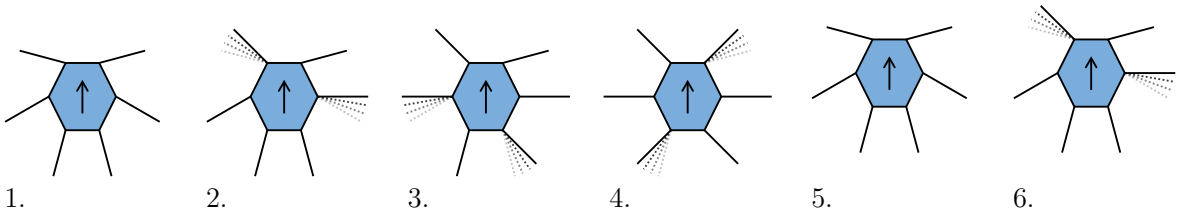


Figure 2: The visualization of the six steps of the tetrapod gait. In step 1., all legs remain in the old footholds. The step 2., the first pair of the legs is reaching new positions. The second pair of the legs is moving to new footholds in step 3. followed by the last two legs in step 4. The steps 5. and 6. are similar to the steps 1. and 2.

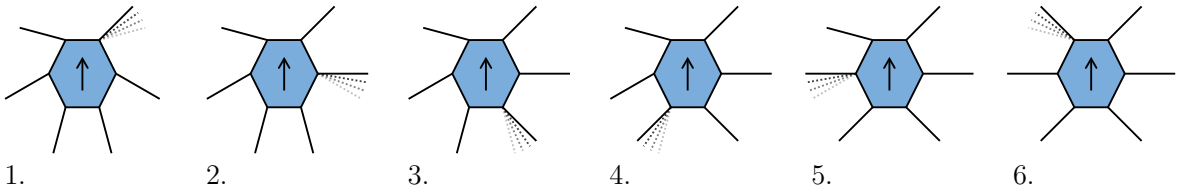
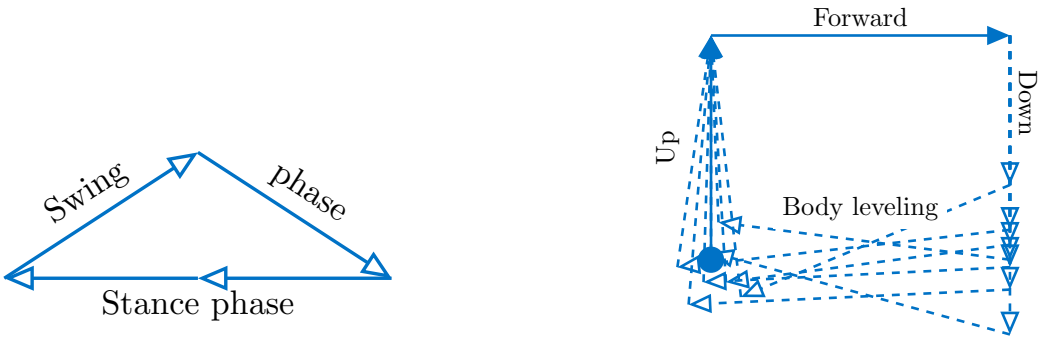


Figure 3: The visualization of the six steps of the pentapod gait. In each step (1.-6.) only a single leg is reaching new foothold, which makes the pentapod gait slow compared to the tripod gait and the tetrapod gait.

non-periodic gaits, for example *free gait* or *wave gait*. *Free gait* use other factors such as external perception of the robot to choose the order of the legs to traverse tough terrains.

2.1 Regular and Adaptive Gait



(a) Regular gait

(b) Adaptive Gait

Figure 4: Regular gait leg trajectory visualized in (a) without any feedback; and several examples of adaptive gait leg trajectories (b). The adaptive gait combines the ground detection during the swing-down phase of the leg with the consecutive body leveling phase which causes the irregular rectangle-like quadrilateral shape of trajectories.

Regarding the ability of the locomotion to traverse rough terrains we can further distinguish two additional types of gaits: *regular gait* for flat terrains and *adaptive gait* for irregular terrains. Legs driven by the ***regular gait*** follow the predefined fixed trajectory. An example of leg endpoint trajectory is shown in Fig. 4a, where the leg movement in the swing phase is divided into two lines, but trajectories of different shapes are also possible. The regular gait assumes flat horizontal surfaces,

therefore all points are strictly defined, which makes this gait type useless for traversing irregular terrain.

On the other hand, the ***adaptive gait*** specifies only certain points of the trajectory, which provides adaptivity of the shape of the trajectory. Even though different trajectory shapes are possible, the authors of [2] decided to splits the leg movement during the swing phase into three lines; up-line (swing up), forward-line (swing forward), and down-line (swing down) resulting in a rectangle-like quadrilateral trajectory shown in Fig. 4b. In this particular case, only upper-points (ending point of the up-line and starting point of the down-line) are defined, ground contact points (starting point of the up-line and ending point of the down-line) are variable and limited only by a minimal reachable point is given by leg the construction. In rough terrains, the position of the ground contact points varies during the hexapod movement because of differences in the terrain heights. Therefore a detection of the foot-strikes is required to stop the leg movement in the swing down phase at the time when the leg reaches the ground contact point.

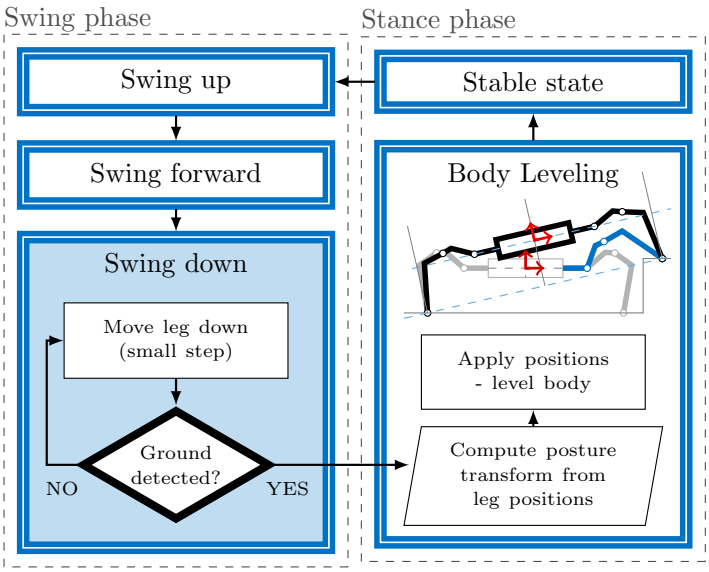


Figure 5: Overview of the gait cycle (stride) together with the visualization of the body leveling.

In the case of the missed foot-strike, the ground reaction force increases, which can result in stability loss, actuators overheat or damaging the robot itself. Therefore the main problem being addressed in this thesis is to provide fast and reliable foot-strike detection using relatively cheap accelerometers. We aim to keep the reliability of the previously deployed force threshold-inspired position (FTP-inspired) control presented by J. Mrva and J. Faigl in [2]. The FTP-inspired control implemented by the authors in the adaptive gait is based on the position error, which is a difference between the expected position of the leg and the real position read from servomotors. The leg motion in the swing down phase is divided into small steps while the current leg position is being read from the servos after the each step.

Whenever leg descending is stopped or slowed by the ground, the position error starts to increase over the motion cycle and if the position error exceeds the given offset the adaptive gait classifies a foot-strike and stops the leg movement. Although the adaptive gait provides a reliable foot-strike detection mechanism, its speed is limited by reading speed of the current position of the servos required at each iteration of the motion cycle. The current servo position can be read only every 16 ms and as long as each motion cycle requires reading from 3 servos, the overall time required for collecting the all needed data increase to 48 ms.

2.1 Regular and Adaptive Gait

The communication delay of the method suggested by J. Mrva and J. Faigl in [2] limits the gait speed and consequently makes the robot driven by this approach relatively slow. Therefore we aim to improve the adaptive gait using inertial data from the accelerometers instead of slow-to-obtain servo positions. In particular, we consider cheap and easy-to-use hexapod platform with 18 Dynamixel AX-12 servos with a single relatively inexpensive accelerometer (ADXL345) attached to each leg. Optionally we add one more inertial measurements unit (namely AHRS unit Xsens MTI-30) mounted on the hexapod body to obtain the absolute body orientation.

Two main issues are related to the considered accelerometer measurements. Firstly, the movement is divided into small discrete steps, which cause vibration. These vibrations along with high acceleration peaks in the sharp corners of the leg trajectory then spread through the whole robot and affect the accelerometer measurements. This noise in measured data makes the foot-strike detection based on inertial measurements more challenging. Therefore, an event detection mechanism is required to correctly detect foot-strike events and maintain the robot attitudes when crawling the irregular terrains. Secondly, the leg position and robot orientation are changing over time and differences in the mounting direction of the accelerometers requires data unification to use one foot-strike detector for all of the hexapod legs.

In addition to problems related to inertial measurements, using machine learning classifiers requires a locomotion controller capable of crawling irregular terrains to collect real data from a rough terrain for learning. Then these classifiers can be deployed in irregular terrains to classify foot-strikes by themselves. Since FTP-inspired controller implemented in the adaptive gait already provides this capability, it is used for both learning data collection and for labeling collected data.

Chapter 3
State of the Art

Two areas of science are mainly related in the topic of the herein studied on-line foot-strike detection. Firstly, in biomedical engineering, requirements for precise foot-strike detection methods arose in the 1960s as a response to first experiments with Functional Electrical Stimulation (FES). The FES replaces or amplifies electrical pulses in the muscles, which helps individual to restore their walking abilities affected by injury of the central motor nervous system (such as "drop foot" problem, Parkinson disease, spinal cord injury, brain trauma, etc.). Time precise distinction of the gait phases is required to improve the efficiency of the FES, Note some authors [4–7] differentiate between more phases than just the stance and swing phase, however the further distinction is based on the position of the toe and heel during leg locomotion; ergo it is specific to a human-like leg structure.

Secondly, in mobile robotics, the advancements initiated the second wave of interest into this topic at the end of 20th century. Progress in mobile robotics makes existence of human-like robots and multi-legged robots possible and the foot-strike detection became required for the correct locomotion, maintaining the humanoid stability [8, 9] and leg sensing data was even used for terrain classification [10]. In the following sections, the approaches are discussed based on two criteria; used sensors and used classification method, followed by brief introductions of the Support Vector Machines (SVM) and the Neural Network (NN) approaches utilized in our proposed solution to the on-line foot-strike detection.

3.1 Suggested sensor combinations

Several sensors and sensor combination have been proposed in the literature to be used in collecting data for foot-strike event classification. A most straightforward approach such as [11] uses only micro-switches attached under a foot. Micro-switches can also serve for a validation of detected foot-strike events as in [12]. Another relatively straightforward proposal uses solely force sensitive resistors (FSR), which changes its electrical resistance based on the applied force. Usually, several FSRs are used to detect heel-load and toe-load, hence one FSR is placed under the heel and several others (namely two in [7] or four in [4]) under a metatarsal head. FSRs are robust, reliable and well-known, and therefore, they are widely used as a reference system in several other approaches (such as [13, 14]). Even though both force sensitive resistors and micro-switch are well-known and reliable, we decided not to use them for several reasons. Firstly, we expect that both above-mentioned sensors would behave rather odd in the sand and generally loose terrain. Secondly, information obtained from micro-switch is strictly limited into two states, hence it is impossible to use that data for terrain classification[1]. Lastly, force sensitive resistors have low accuracy, even though using an array of FSRs can improve its performance [15], and therefore, they are not used for terrain classification, even though K. Suwanratchatamanee and M. Matsumoto in [9] used the triplet of FSRs to determine slope orientation under humanoid robot legs.

Another widely used sensors are accelerometers, either just one or several, all uniaxial, biaxial and triaxial or even triple of uniaxial accelerometers. Using accelerometers provides more variability such as placement, which is not limited under foot-contact part of leg as in cases of FSR and micro-switches. Generally speaking, accelerometers can bring more information, for example they can be

[1] Terrain classification is one of the branches studied in our lab, hence sensors allowing terrain classification are preferred for its possible future use.

used for determination of shank orientation [16, 17] or even determine the whole kinematics chain of the human leg [14, 18] used for the foot-strike event classification, which would be impossible using FSR or micro-switches. Last but not least, accelerometer can be used in the terrain classification for both legged robots [10] and wheeled robots [19], for detection of stair ascend/descend and ground slope (upslope, downslope, ground plane) of human subject [17] as well.

Other possible sensor equipment is a combination of force sensitive resistors (FSRs) on foot and several accelerometers on various parts of the leg [5, 6]. Such a combination brings more information about the leg movement than using only FSRs, but as it is mentioned above, using FSRs cannot easily provide any new data for terrain classification, which makes FSRs surplus for future use in our lab. It is worth mentioning that most of the approaches use FSRs as a reference for other sensors used for foot-strike detection. Beside FSR, force sensitive floor plates [7], visual movement image [20], camera-based position capture system [5], and micro-switches [12] have been used to verify foot-strike event detection approaches. Also note that force/torque sensors are used for terrain classification [8], and that approach inspired by the force position threshold is used for a foot-strike detection in the groundwork [2].

3.2 Existing classification methods

Several groups of methods have been developed for foot-strike detection or generally gait phase detection. Firstly, rule-based detection algorithms usually with a finite-state-machine representation of the gait defined with manually set parameters for transitions between the states have been used by a significant number of authors. Secondly, using artificial neural networks (NN) has been proposed by several authors and lastly Support Vector Machines (SVM) have been used as well.

3.2.1 Rule-based detections algorithms

Rule-based detection algorithms generally require a set of rules used for classifications of the collected data. There are two main approaches implemented for the rule-based detection. Cycle-like structure of the gait is presented by J. K. Lee and E. J. Park in [12] and by H. Jonghee et al. in [20]. In these works, only two states are defined, ergo transitions between all states of the gait are valid. J. K. Lee and E. J. Park in [12] have been using local maxima and local minima of the angular velocity and angular acceleration to define conditions for the transition between the foot-end contact (beginning of the swing phase) and initial contact states (end of the swing phase), which is similar to foot-strike detection. Similarly H. Jonghee et al. have been looking for peaks in the horizontal and vertical acceleration to distinguish between swing and stance phase in [20].

Beside the cycle-like structure, the finite state machine gait representation have been suggested by I. P. I. Pappas et al. in [5], later in [6], A. T. M. Willemsen et al. in [21] and by Milica Djurić in [14]. All the above-mentioned authors differentiate more than two gait phases. Gait phases are expressed as a finite state machine in which states correspond with the gait phases, and possible transition between the states (gait phases) are defined based on human gait. I. P. I. Pappas et al. in [5] and later improved in [6] uses three FSRs and gyroscope to distinguish between five gait states supporting some pathological features of the gait such as shuffle or shamble. Their approach has been tested on various surfaces. Milica Djurić in [14] reconstructs the position of each leg part using up to six triaxial accelerometers. His approach uses the set of four conditions derived from the inertial measurement to determine the gait phase. In [21] A. T. M. Willemsen et al. use only four uniaxial accelerometers (basically two biaxial accelerometers) and similarly to I. P. I. Pappas et al. differs between five gait phases.

Most of the algorithms mentioned in this subsection require either combination of FSRs and accelerometers [5, 6] or require multiple accelerometers [21] on a different part of the leg, which would

be challenging for our platform and would require different classifiers for particular n-pod gaits. Reconstruction of the leg in [14] a priori assumes that relative position of the leg parts is related to the gait phase, which might be valid for human subjects using own sensors to move their legs, but it cannot be expected from our robotic platform.

3.2.2 Artificial Neural Networks

Another group of authors using both sole accelerometers, a combination of accelerometers and FSRs, and even solely FSRs bets on using artificial neural networks (NN). In [7] Adam Miller proposed using simple multilayer neural network for classification of foot-contact and foot-off (swing and stance phase) using data from two FSR under heel and toe metatarsal bone. Similarly, in [13] N. Mijialović et al. propose to use a two-layer neural network with four input neuron, 15 neurons in each of hidden layers and one output neuron designed to classify swing and stance phases based on the data from 2 triaxial accelerometers[2]. An almost identical approach was suggested independently by Y. Schimada et al. in [22], where the authors developed a neural network to distinguish between two gait phases as well. The proposed neural network uses only measurements from one biaxial accelerometer located on the thigh of human subjects. S. Došen et al. present neural network classifier [18] as a part of the optimal walking gait controller using a combination of four FSRs and four biaxial accelerometers on foot, shank, thigh, and even pelvis. The authors construct an input vector consisting of the current accelerometer dataset, two sets of delayed accelerometer data, the current set of FSRs data, and two sets of delayed FSRs data. This huge input vector is being fed into neural network consisting of one hidden layer with ten neurons and one output neuron. An interesting approach using an Adaptive logic network (neural network that changes the size of the hidden layer when needed) has been developed by R. Williamson and B. J. Andrews in [16]. This approach is obtaining data from three uniaxial accelerometers located on a knee of the human subject. Another approach [19] suggested by S. Otte et al. is using recurrent neural networks (RNNs), more precisely the Dynamic Cortex Memories (DCMs are improved long-short-term memory NNs), and vibration data of wheeled robot to classify 14 types of mainly outdoor terrain.

Even though some approaches uses previously rejected sensors (i. e., the FSRs or micro-switches) [7] or requires a combination of rejected sensors and a large number of accelerometers [13], two approaches look promising. Y. Schimada et al. propose in [22] to use only two triaxial[2]accelerometers relatively close to each other and R. Williamson and B. J. Andrews propose in [16] to use data from three uniaxial accelerometers located only a few centimeters apart, which is in both cases similar to our suggestion of using only one triaxial accelerometer per a single leg of the robot.

3.2.3 Support Vector Machine

Last but not least group of approaches considers Support Vector Machine (SVM) for gait phase distinction, terrain classification, and slope classification. The authors of [4] collect data from four FSRs (two under heel and two under first and fifth metatarsal head) to detect five gait phases using SVM. Similarly, L. Hong-Yin and T. Kai-Yu use data from three accelerometers to differentiate between the swing and stance phases utilizing the SVM classifier in [17]. Regarding the gait phase classification, F. L. C. Bermudez et al. deployed SVM terrain classifier for legged robot in [10]. The proposed classifier uses various data from onboard sensors (inertial measurement unit, motor control, and magnetic encoders) and it is capable of distinguishing three different types of terrain. SVM method has been also used for terrain classification by K. Walas et al. in [8]. The authors propose collecting force and torque data followed by the Fast Fourier Transformation and the Discrete Wavelet Transformation to identify five different types of mainly indoor-like terrains.

[2] Authors use only two most significant axes from each of the triaxial accelerometers.

Table 1: Summary of the State of the Art Approaches

	Gait Phases	Accelerometer	FSR	Detection method
A. T. M. Willemsen et al. [21]	5	4 uniaxial	—	rule-based (FSM)[3]
I. P. I. Pappas et al. [5]	5	1 uniaxial	3	rule-based (FSM)[4]
I. P. I. Pappas et al. [6]	5	1 uniaxial	3	rule-based (FSM)[4]
Lee and Park [12]	2	1 triaxial	—	rule-based[3]
Williamson and Andrews [16]	5	3 uniaxial	—	neural network – ALN[4]
B. Huang et al. [4]	5	—	4	SVM
Adam Miller [7]	2	—	2	neural network
N. Mijailović et al. [13]	2	2 triaxial[5]	—	neural network[4]
H. Jonghee et al. [20]	2	1 triaxial	—	rule-based
Y. Shimada et al. [22]	2	1 biaxial	—	neural network
Milici Djurić [14]	7	3 triaxial	—	rule-base (FSM)[4]
Došen and Popović [18]	–	4 biaxial	4	neural network

Even though some of the methods again uses previously rejected sensors (i. e., the FSRs) [4, 10], there are approaches supporting our intentions of using an accelerometer to provide data for the SVM classifier [8, 17]. In the following section, we discuss Support Vector Machines and used types of Neural Networks.

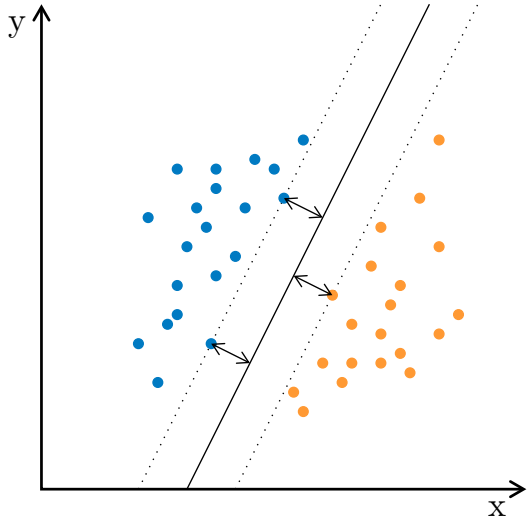
3.3 Support Vector Machine

The Support Vector Machines (SVMs) are set of methods of supervised machine learning presented by Vladimir Naumovich Vapnik and Hava Siegelmann in 1963 [23] and extended during the following years by other authors [24]. The SVM can be used for the classification [25], regression [26] or other tasks such as the outliers detection [27]. However, the SVMs are mostly used for the classification.

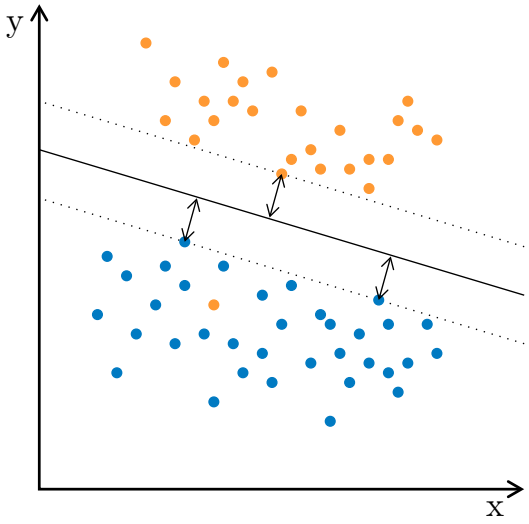
In the SVM, each input data vector of the dimension n is represented as a point (vector) in the n-dimensional features space. The SVM algorithm then finds the hyperplane in the n-dimensional feature space, which divides points into two classes. We called the training data linearly separable if we can find two parallel hyperplanes that separates the classes so that the distance between these two hyperplanes is the longest possible. Any hyper-plane in the area bounded by these hyper-planes (so-called "margin") separates the input data into appropriate classes, therefore additional selection is required. V. N. Vapnik and H. Sieggelmann proposed to use hyperplane, which distance from the both bound hyper-planes is the longest possible, ergo the hyperplane in the middle of the "margin" area. An example of the maximum-margin hyperplane, the "margin" area and the bound hyper-planes are visualized in two-dimensional space on the linearly separable data in Fig. 6a. The margin of the maximum-margin hyperplane in the case of the linearly separable data is so-called hard-margin.

There are also data set, for which it is impossible to find maximum-margin hyperplane. In this cases, we call the data linearly non-separable, and other approaches are used. For the linearly non-separable input data, the outliers in each class are ignored, and the hyperplane with the maximal margin is used to differentiate between two classes. We call the margin of the linearly non-separable data the soft-margin. The hard-margin is always preferred by the SVM over the soft-margin, even though the soft-margin of linearly separable data set is bigger than the hard-margin, the points near the different class (possible outliers) are not ignored at the expense of the lower hard-margin. An

[3] Authors used micro-switches for validation of proposed method.
[4] Authors used force sensitive resistors for validation of proposed method.
[5] Authors use only two most significant axis from each of the triaxial accelerometers.



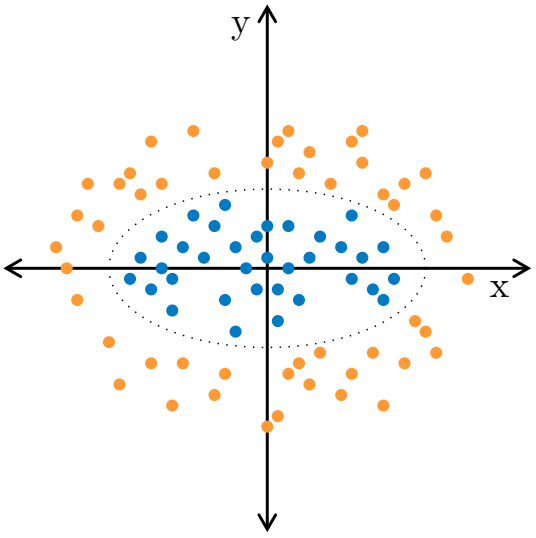
(a) An example of linearly separable two-dimensional input data. The class A (blue) and the class B (orange) can be divided using hard-margin hyper-plane (full line) in the middle of the margin area bounded by dotted lines. The distances to the closest samples from both classes are marked by double-head arrows.



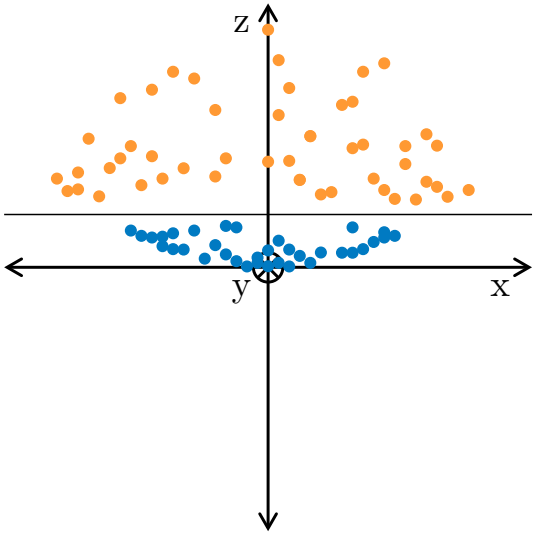
(b) An example of the linearly non-separable two-dimensional input data. The class A (blue) and the class B (orange) cannot be separated using the hard-margin hyper-plane because of the outlaying class B sample. Therefore the soft-margin hyper-plane (full line) is used and visualized in the center of the margin bounded by dotted lines.

Figure 6: An example of the linearly separable data (a) and linearly non-separable data (b) with the hard-margin and soft-margin respectively marked.

example of the linearly non-separable data is visualized in the Fig. 6b.



(a) An example of the raw input data in two-dimensional space. The class A (blue) is bounded by an ellipse, therefore the polynomial kernel function can be used to separate the class B (blue).



(b) The data raw data (a) with the feature z added using the polynomial kernel function $z(x, y) = x^2 + 0.25y^2$. The three-dimensional space is projected into the x-z plane projection.

Figure 7: The visual comparison of the raw input data (a) and the data after adding new feature using the polynomial kernel function (b).

In the 1992, the B. E. Boser et al. [24] presented the kernel functions to improve the SVM based on the work of M. A. Aizerman et al. [28]. The proposing of the kernel functions transforms the feature space. The kernel function transformation can be nonlinear and can increase the dimension of the feature space. The linear, the polynomial, the sigmoid, the hyperbolic tangent, and the radial basis functions are usually used as a kernel function, even though defining custom kernel function is also possible. In the example (Fig. 7) of nonlinearly separable data, the polynomial kernel function $z(x, y) = x^2 + 0.25y^2$ allows using the hyperplane for the classification.

3.4 Neural Network

The neural networks are another set of methods of the supervised machine learning methods that are inspired by biological neural networks. The original idea of using neural network was proposed by Warren McCulloch and Walter Pitts in [29], but the first simplified neural network called perceptron was developed in the 1960's by Frank Rosenblatt. The further improvement resulted in the first multi-layer neural network proposed by Alexey G. Ivakhnenko et al. in 1965.

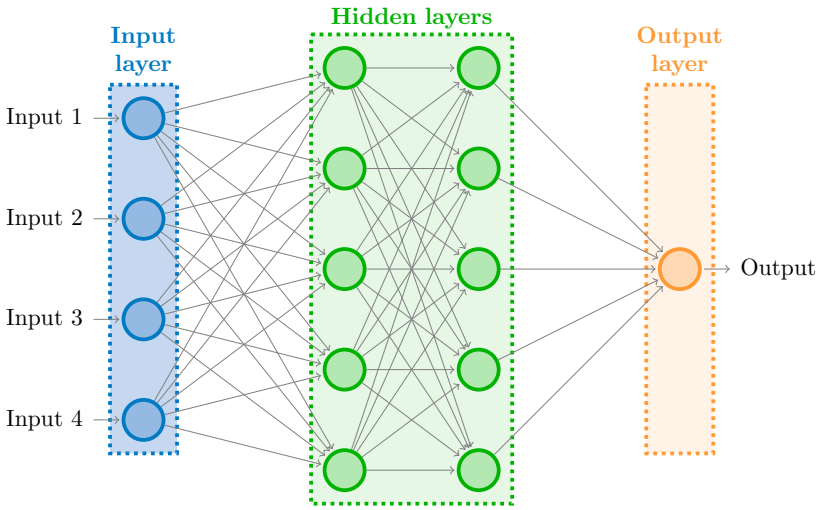


Figure 8: An example of the neural network structure with four neurons in the input layer (blue), five neurons in each of the hidden layers (green) and a single output neuron (orange).

Neural networks consist of individual units called neurons. Each neuron has a given number of inputs $\mathbf{x}$, one output, activation function and bias b. Neurons are connected via edges, and they are usually grouped into layers as visualized in Fig. 8. Each edge starts from the output of the neuron and ends in the input of the other neuron. The activation function of the neuron takes as an argument sum of the weighted input signals z,

$$z = \sum_j w_j x_j = \mathbf{w}.\mathbf{x}, \tag{1}$$

where the vector of weights $\mathbf{w}$ represents a relevance of the vector of the input $\mathbf{x}$. The vector of the weights changes during the learning process, which is described further.

Based on the activation function, we can distinguish between the *perceptron* and the *neuron*. The perceptron uses the *step function*:

$$s(z) = \begin{cases} 0 & z \leq \text{bias} \\ 1 & z > \text{bias} \end{cases}, \tag{2}$$

where the bias is the threshold value. On the other hand, the neuron uses the group of activation

functions called *sigmoid functions* such as hyperbolic tangent function (3), arctangent function (4) and the logistic function (5), that is so called sigmoid function.

$$\sigma(z) = \tanh(z) \qquad (3) \qquad \sigma(z) = \arctan(z) \qquad (4) \qquad \sigma(z) = \frac{1}{1+e^{-z}} \qquad (5)$$

The connections between the neuron layer define the types of the neural networks. Three main neural network types relevant for this thesis are further discussed; multi-layer neural networks, long-short-term memory neural networks, and the recurrent neural networks.

The multi-layers neural networks are commonly used for various tasks [30]. They consist of the input layer, numerous hidden layers, and the output layer as visualized in Fig. 8. The number of neurons in the input layer varies based on the number of the features in the input data, and the number of neurons in the output layer depends on the number of classes. The number of hidden layers and the number of the neurons in the hidden layers depends on the solved problem.

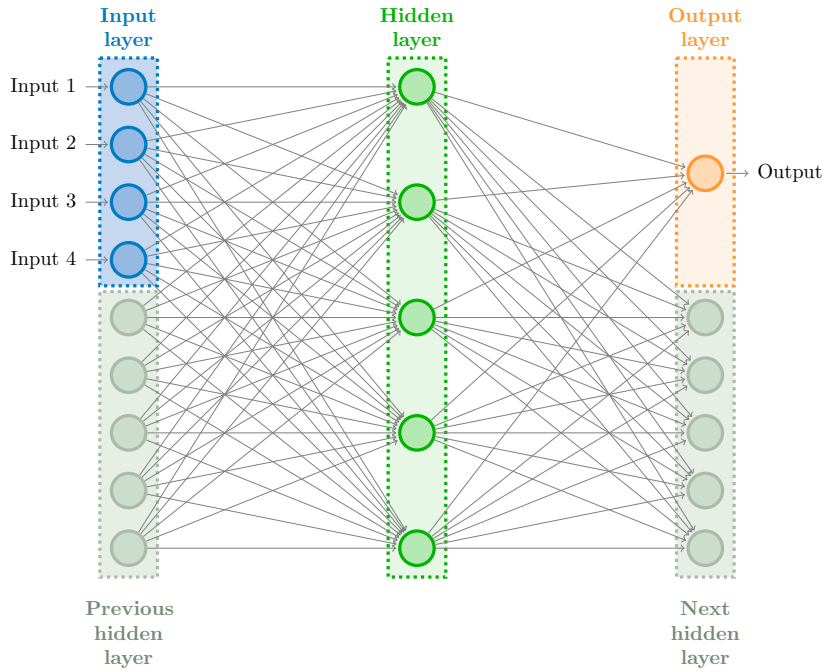


Figure 9: An example of the recurrent neural network structure with four neurons in the input layer (blue), five neurons in each of the hidden layers (green), and a single output neuron (orange). The outputs of the hidden layer from the previous sample (gray, left) are used by the hidden layer, therefore the previous hidden layer is visualized under the input layer. The outputs of the current hidden layer are used by the RNN for the next sample, hence the following hidden layer (gray, right) for the next sample is visualized under the output layer.

The Recurrent Neural Networks (RNNs) are neural networks, which current state is influenced by its previous states. They are usually used for processing of stream-like data, where the next state is affected by the previous state or states; e. g. speech recognition, language modeling, translation, image captioning, etc. [31]. The information transition between each state is in the RNNs done by connecting the output of the hidden layers from the previous states to the input of the hidden layers of the current state as visualized in Fig. 9. The dimension of the input, output, the number of the hidden layers and the number of neurons in the hidden layers are set similarly to the multi-layer NNs.

The RNNs are capable of the good performance on the data, where the current output depends on the relatively recent previous inputs, whether they behave poorly on the data with long-term relations [32].

The Long-Short-Term Memory (LSTM) NNs are modified RNN, which improve their ability to learn the long-term relations. The LSTM was firstly introduced in 1997 by Sepp Hochreiter and Jürgen Schmidhuber [33]. Similarly to the RNNs, also in the LSTMs the previous state propagates to the current state. Besides a long-term memory connection is implemented to transfer additional information over time.

Regardless the structure of the neural network, the learning process is required. Even though D. O. Hebb in 1949 proposed unsupervised learning method called *Hebbian learning* [34]; the backpropagation supervised learning method proposed by Paul Werbos in his thesis [35] is predominantly used for learning neural networks. The backpropagation modifies the weights of the neurons based on the classification error. The classification error is *backpropagated* from the output of the neural network through the last hidden layer way up to the first hidden layer. During the backpropagation, the weights of the neuron inputs are modified accordingly. The backpropagation is then repeated during the learning process. Certain neural network types require modified backpropagation process, for example, RNNs uses backpropagation through time (BPTT) [36].

Chapter 4
Used Hardware and Platform

In this thesis, we propose to improve the existing platform presented by J. Mrva and J. Faigl in [2] that is composed only cheap and easy-to-use hexapod platform and a central control unit. We have extended this platform by adding six three-axial accelerometers Analog Devices ADXL345 one on each leg and central Xsens MTI-30 three-axial inertial measurements unit is shown in Fig. 10. This chapter describes the hardware used in our work. Firstly, hardware communication is introduced followed by a brief description of the used platform and central control unit. In the middle of the chapter, the proposed accelerometers are characterized and their measure modes and other features are presented. Lastly, the multiplexer printed circuit board and central inertial unit are described.

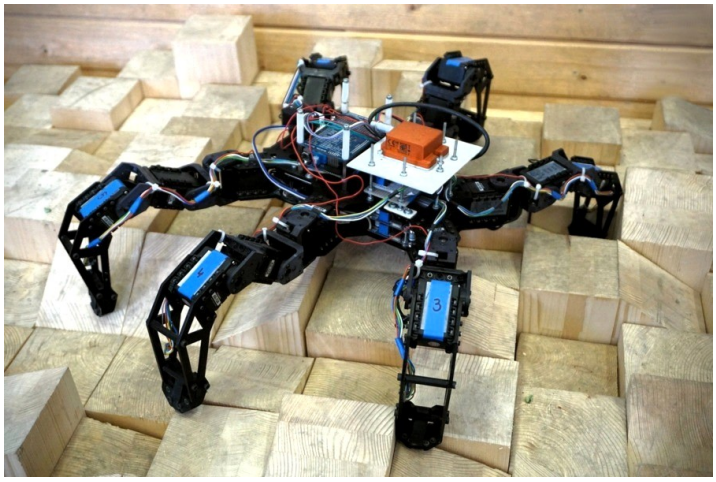


Figure 10: Overview of the used platform with all additional equipments.

4.1 Hardware Connections

Three main communication interfaces are used for the communication; I^2C (Inter-Integrated Circuit), USB (Universal Serial Bus) and UART (Universal Asynchronous Receiver-Transmitter). The I^2C is used for collecting data from the accelerometers, USB is used for communication with Xsens and the UART-to-USB converter is used for sending instructions to servos.

All accelerometer I^2C communication interfaces are connected to hand-designed multiplexer printed circuit. Signals (SDA - serial data and SCL - serial communication clock) are multiplexed into single I^2C bus, hence three multiplex address pins (ADD) are used to choose the particular accelerometer. The address pins for the multiplexer and single I^2C bus is attached through the Shifter Shield extension board to the Odroid central control computer of the robot. Beside the I^2C communication interface, each accelerometer is connected directly to the Shifter Shield via its INT (interrupt pin) to transfer the interrupt signal to Odroid. All pins from Odroid itself are remapped to the Shifter Shield.

Servo motors are commanded via UART bus connected to UART-to-USB converter, which is linked via USB to Odroid. Xsens inertial unit is directly attached to Odroid via USB. The verall

communication structure of the used hardware is visualized in Fig. 11. The power source (Li-Pol battery or adapter) is connected through a pair of switches to both AX/MX Power Hub (inside the UART-to-USB converter) and Odroid central control unit.

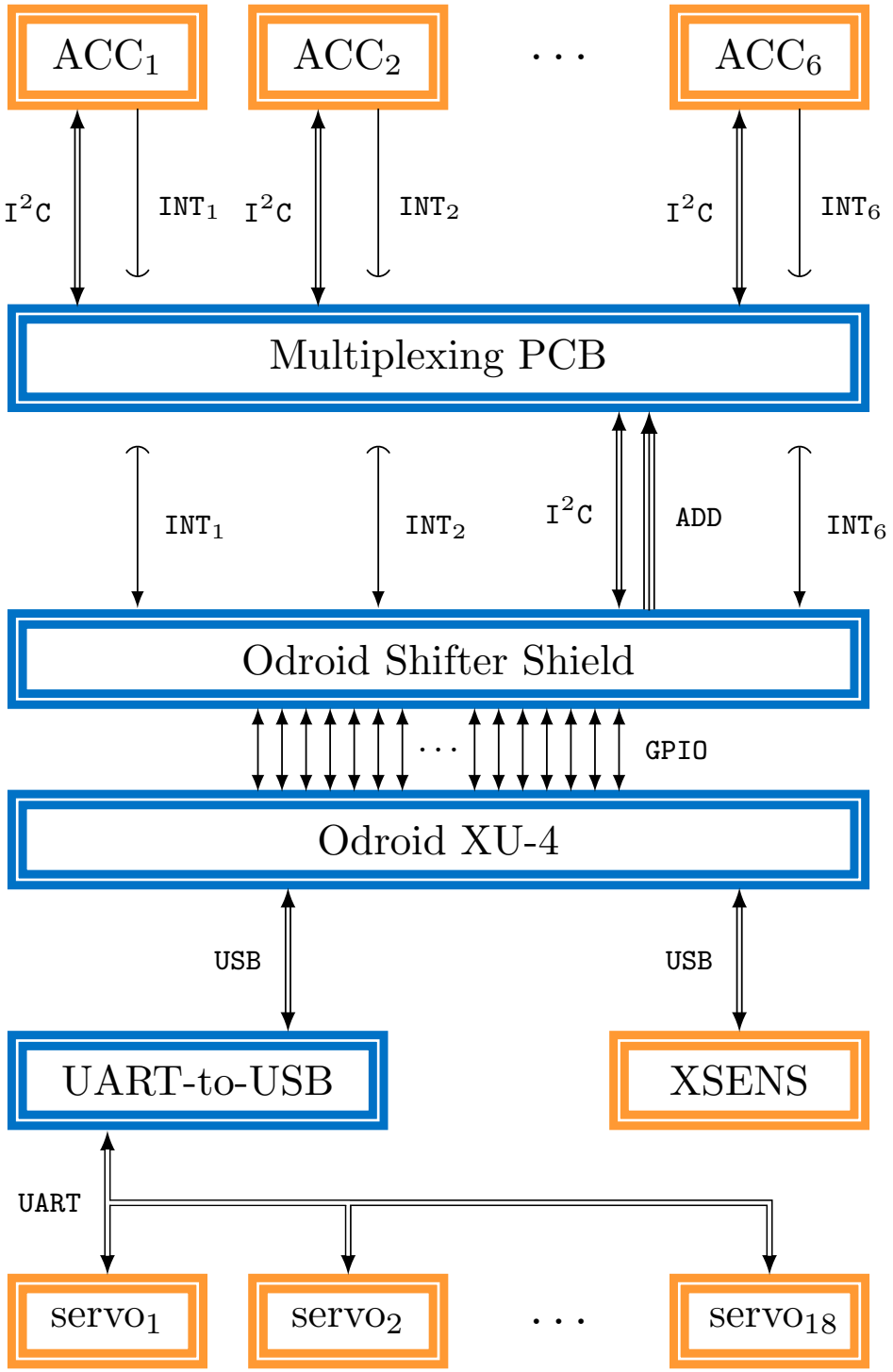


Figure 11: Overview of the hardware connection between the main components of the hexapod platform. Actuators and sensors (orange) and electronic parts (blue) are connected via buses (double lines) or single wire (single line).

4.2 Hexapod body

The used platform consists of 18 Dynamixel servos connected to UART-to-USB converter, plastic body plates, and frames. Each of six hexapod legs is composed of several plastic parts and three servos which make three degrees of freedom (DoF) per each leg. The platform has two separated power circuits. One for servomotors and the second one for the central control unit allowing turning off overheated servos while keeping the central control unit running. Additional holes for bolts fastening accelerometers to legs and other components screwed to the body were drilled into the plastic body frames and plates.

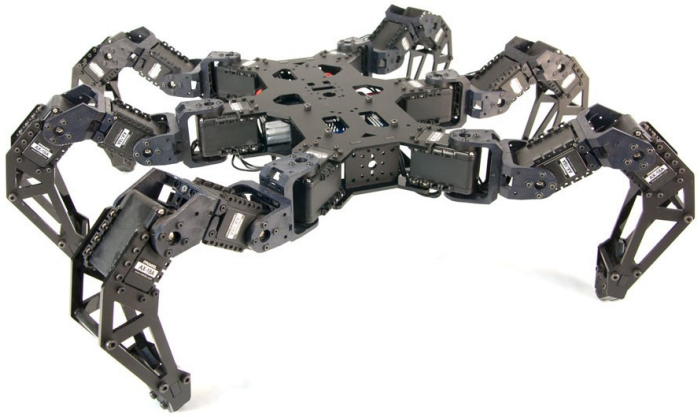


Figure 12: Overview of used platform .

4.2.1 Hexapod Leg

Each hexapod leg is composed of three links and three servomotors. Links and servos are called from the body to the foot: *coxa*, *femur*, and *thibia* as shown in Fig. 13. Values read from the actuators are further referred as θ_c for the *coxa* servomotor, θ_f for the *femur* servomotor, and θ_t for the *thibia*.

The *Coxa* joint allows leg movement in a plane parallel to the body plane, while the *femur* and *thibia* joints allow movement in the plane that is perpendicular to the body plane and on the position of the *coxa* joint. This construction feature is used in the adaptive gait [2] where the movement of the *coxa* servo and movement of the pair of *femur* and *thibia* servos are split into sub-phases (see Section 2.1) such that whenever a leg is moving vertically, mainly the *femur* servo is moving, while the *thibia* servo rotates minimally. Therefore reading position from single servo is required by the adaptive gait [2].

4.2.2 Servomotors

The utilized servomotors (Dynamixel AX-12A) communicate with the central control unit via UART communication interface, which provides half duplex asynchronous serial communication. The servos can operate in two modes; *joint mode* and the *wheel mode* suitable for wheeled or tracked robots. In both cases, the resolution of the servo is limited to 1024 units per a single rotation, which gives the resolution 0.35°. Moreover the joint mode limits maximum rotation to 300° range [37].

The used servos allow setting the desired position every 1 ms, however, reading the current servo position takes about 16 ms. The period is caused by the communication delay between the request for

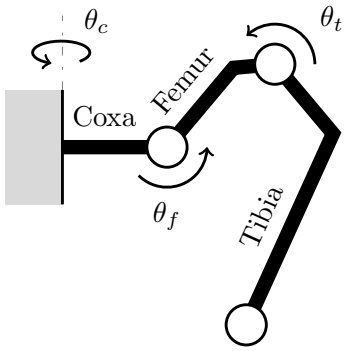


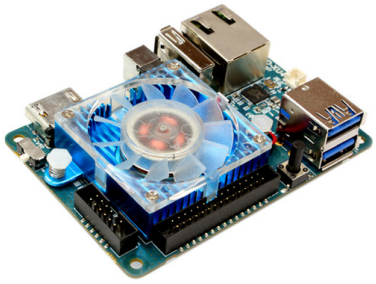
Figure 13: Hexapod leg scheme with leg parts, joints, and joint angles labeled.

Figure 14: Front and rear view of used servos Dynamixel AX-12A.

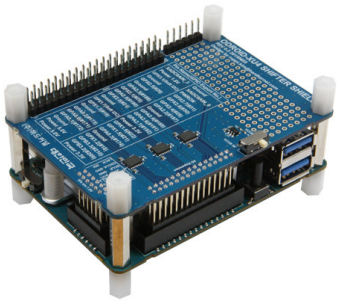
the current position and servo response. The groundwork approach used in the adaptive gait requires luckily only reading position from one servo (namely *femur*) which turns out to 48 ms for one iteration of the tripod gait cycle. Even though the number of readings was reduced to the minimum value, this delay still represents a significant bottleneck in the FTP-inspired approach [2], which is addressed in this thesis by the proposed foot-strike detection using accelerometer measurements.

4.3 Central Control Unit

The proposed approach uses the Odroid XU-4 embedded computer as the central control unit. The Odroid XU-4 is equipped with 2 GHz ARM CORTEX A7 octa-core processor (Samsung Exynos5422) accompanied with 2 GB RAM and 16 GB eMMC5.0 HS400 Flash storage with Ubuntu 16.04.2 LTS. Additionally, an active fan cooling unit has been chosen over passive tall-blue-heat-sink cooler to allow placing XU-4 Shifter Shield over the Odroid itself. We have decided extends the Odroid XU-4 by adding the Shifter Shield extension board, which enables using 3.3 V or 5 V logic while the general purpose input output (GPIO) pins of the Odroid natively operate at 1.8 V [38]. This voltage amplification is necessary because the supply voltage of the used accelerometers is specified in the range from 2.0 V to 3.6 V [39]. Even though the plain Odroid XU-4 allows using up to six pairs of pins as six independent $\mathtt{I^2C}$, only two pairs of pins are usable for $\mathtt{I^2C}$ by adding the Shifter Shield necessary for powering the accelerometers [38].



(a) Central processing unit Odroid XU-4 with fan cooling unit.[6]

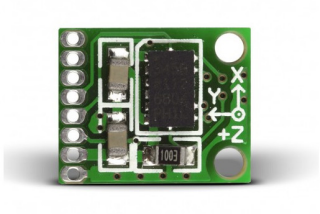


(b) Central processing unit Odroid XU-4 covered by Shifter Shield extension board.[7]

Figure 15: Central processing unit Odroid XU-4 with and without the Shifter Shield extension board.

4.4 Accelerometer[8]

The used accelerometer ADXL345 Analog Devices soldered on two different printed circuit board (PCB) (as seen in Fig. 16a and Fig. 16b) is small, high resolution, triaxial accelerometer with the range up to $\pm 16\ g$. Accelerometers are nowadays used in various applications, therefore ADXL345 is customizable to suit the specific requirements of various applications. The standard *continual* measurement mode as well as *single-tap* and *double-tap* modes, *free-fall* and *activity/inactivity* sensing modes are provided at the hardware level directly by the ADXL345 chip and described in Section 4.4.1. These functions may optionally be mapped on one or two interrupt output pins (visible in the middle of the PCB in Fig. 16b). Beside the above-mentioned modes, the ADXL345 accelerometer provides 32-level first in, first out (FIFO) buffer to store measured values and reduce digital interface usage. The FIFO buffer modes are presented in Section 4.4.2. The ADXL345 supports two digital interfaces; `SPI` (Serial Peripheral Interface) both 3-wire and 4-wire and $\mathtt{I^2C}$ further discussed in Section 4.4.3.



(a) An older version of the used ADXL345 microchip in the ready-to-use board by MikroElektronika.[9]

(b) A newer version of the used ADXL345 microchip in the ready-to-use board by SparkFun.[10]

Figure 16: Two variants of the printed circuit boards (PCB) with the ADXL345 accelerometer chip.

[6] Source: `http://www.hardkernel.com/main/products/prdt_info.php`
[7] Source: `https://wiki.odroid.com/accessory/add-on_boards/xu4_shift_shield`
[8] Most of the information is derived from the ADXL345 datasheet [39].
[9] Source: `https://www.mikroe.com/accel-spi-board`
[10] Source: `https://www.sparkfun.com/products/9836`

4.4.1 Measure Modes of Accelerometer

In this section, we further describe the ADXL345 measure modes with the emphasis on the modes used in our approach (i.e. *stream* and *single-tap*) along with general settings which share all measure modes. Two parameters of the measurement are same no matter what measure mode is chosen; the measure range and the measurement resolution. Four acceleration ranges $\pm 2\ g$, $\pm 4\ g$, $\pm 8\ g$ and $\pm 16\ g$ are possible, and two different output resolutions are implemented; 10-bit resolution and the full resolution. The 10-bit resolution keeps the length of the measured data, but vary in acceleration represented by least significant bit for different ranges (from $3.9\ \mathrm{mg/LSB}$ for $\pm 2\ g$ to $31.2\ \mathrm{mg/LSB}$ for $\pm 16\ g$). On the other hand, the full resolution keeps the value represented by the least significant bit ($4\ \mathrm{mg/LSB}$) same for all ranges and vary in the length of the measured data. Those and every other setting made to customize measurements are stored in series of registers provide by the ADXL345.

Stream mode continuously provides the measured data at the given measure rate. Two measure rates are implemented: standard data mode (from 6.25 Hz to 3200 Hz) for regular operation of the accelerometer; and low power mode (from 12.5 Hz to 400 Hz) for power saving. In our proposed method, only the standard data mode is used.

Single-tap mode collects data continuously, but only if given conditions are met, the measured data are stored in FIFO for further processing. Two parameters specifying the condition for the single-tap are the *threshold* and the *maximum duration*. The *threshold* represents a minimal absolute value of the acceleration that is required for the event to be considered as a possible single tap. Exceeding the *threshold* value in any of the axis starts a countdown. If the acceleration is higher than the given threshold for a shorter time than the *maximum duration*, an event is classified as *single-tap*, and the interrupt pin is set to high (assuming that *single-tap* is enabled). An example of the single-tap even is visualized in Fig. 17 with both mentioned parameters labeled.

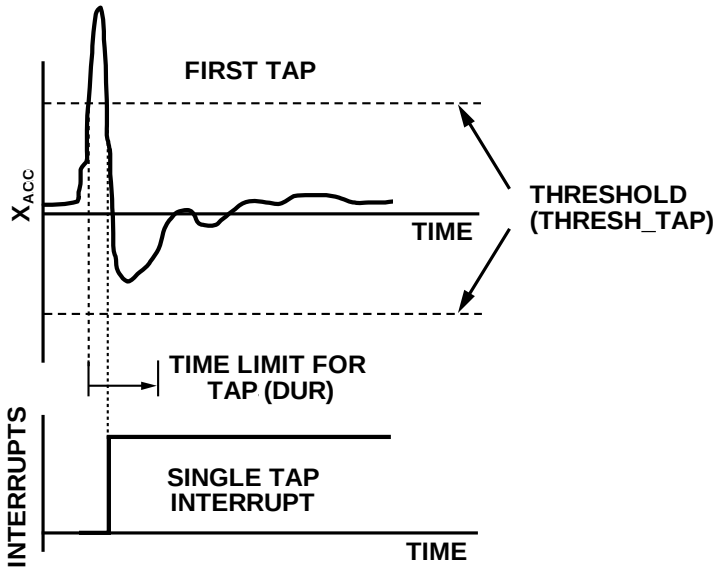


Figure 17: Example of the interrupt triggered by x-axis with settable parameters.[11]

The threshold value is stored in two-byte `THRESH_TAP` register with the scale factor $62.5\ \mathrm{mg/LSB}$ and the maximum duration is saved in the `DUR` register with the scale factor $625\ \mu s/\mathrm{LSB}$. In both cases, setting the value of `0x00` may result in undesirable behavior [39]. The single-tap feature itself and any other features are enabled in the one-byte `INT_ENABLE` register and mapped to interrupt pins by one-byte `INT_MAP` register. Notice that it is necessary to set the value of `0x00` in the `Latent` register to disable the double-tap mode.

Beside the *single-tap* and *stream* modes used in our approach, the accelerometer ADXL345 pro-

vides the *double-tap* detection, *activity* and *inactivity* detection, *free-fall* sensing, *watermark* bit, and *overrun* bit. The *double-tap* mode allows to detect double-tap events defined by the same parameters as the single-tap mode but with additional parameters defining the minimum time between taps and the maximum waiting time for the second tap. The *activity* sensing mode provides a detection of the acceleration activity based on a given minimum acceleration. Similarly to the activity sensing mode, the *inactivity* sensing mode provides detection of the inactivity base on the given maximal acceleration and minimal time of acceleration. The *free-fall* sensing mode allows detecting free falls based on the threshold value and the minimum time of the free fall. Both the *watermark* bit and *overrun* bit are related to the FIFO buffer. The watermark bit has the value of 1 whenever the number of stored samples in FIFO exceeds a given number, and the value of the overrun bit determines whether data in the FIFO registers have been replaced.

4.4.2 FIFO Modes of Accelerometers

In addition to the measure modes, the accelerometer ADXL345 provides an embedded 32-level FIFO (first in, first out) buffer. Similarly to the measurement modes, the FIFO buffer can operate in various modes, and it can be disabled as well. All the settings related to the FIFO are stored in the one-byte `FIFO_CTL` register shown in Tab. 2. Two the most upper bits (`D7,D6`) are used for selecting the FIFO mode, the trigger bit (`D5`) is used in the *trigger mode* described further and purpose of the bits (`D4-D0`) varies for different FIFO modes.

Table 2: Functions of the `FIFO_CTL` register bits.

	D7	D6	D5	D4	D3	D2	D1	D0
	Mode		**Trigger**	**Sample bits**				
Bypass mode	0	0	*none*	*none*				
FIFO mode	0	1	*none*	the number of samples for the watermark bit				
Stream mode	1	0	*none*	the number of samples for the watermark bit				
Trigger mode	1	1	use `INT(1+D5)`	the number of samples before the interrupt				

The ***bypass mode*** bypasses (disables) the FIFO buffer, which therefore remains empty. In this mode, the values of bits `D4-D0` in the `FIFO_CTL` register are ignored.

In the ***FIFO mode***, the accelerometer collects data and stores them in the FIFO buffer. If the number of samples in the FIFO buffer exceeds the value of the sample bits, the watermark interrupt is set to 1. FIFO continues saving measured data until it is full and then stops. Note that the accelerometer stay operational, therefore other features such as single or double tap can be used regardless FIFO is filled.

The ***stream mode*** is similar to the FIFO mode; the usage of the sample bits remains the same as the values stored in `D4-D0` determine the number of samples required to set the watermark interrupt bit to 1. The only difference between this two modes is in handling cases when the FIFO buffer is full. The previously mentioned FIFO mode stops collecting data after filling the FIFO buffer, but in the stream mode, older data are replaced by newer.

In the ***trigger mode***, the accelerometer waits for an interrupt to occur on either `INT1` or `INT2`. If the trigger bit (`D5`) in the `FIFO_CTL` register is set to `0`, `INT1` is used to trigger the trigger mode, et vice versa (setting `1` in the trigger bit causes the trigger mode to be triggered on `INT2`). After the interrupt occurrence, the FIFO buffer stores the last n samples defined in the sample bits and continues to collect samples until the FIFO buffer is full. In the trigger mode, collecting new data is stopped when the FIFO buffer is filled.

4.4.3 Communication with Accelerometer

The ADXL345 accelerometer provides two communication interfaces; SPI and I²C. Even though SPI provides higher communication speed, I²C has been chosen over SPI for two main reasons. Firstly, I²C requires only two wires (SDA and SCL as shown in Fig. 18) to transfer the data. The SPI, on the other hand, requires three (SDIO, SCLK and $\overline{CS}$) or four (SDO, SDIO, SCLK and $\overline{CS}$) wires. Choosing the SPI would therefore significantly decrease the number of the GPIO pins usable in the future. Secondly, the Odroid supports natively I²C communication interface, which makes implementation easier and reduces communication errors. The number of wires required for different communication interfaces is shown in Tab. 3.

Table 3: Number of wires required by communication interfaces.

Communication interface	**Wires per accelerometer**		**Total number of wires required**	
	without interrupt	with interrupts	with interrupts	with interrupts and address
I²C	2	3	18	11
SPI (3-wires)	3	4	24	—
SPI (4-wires)	4	5	30	—

The chosen I²C is the synchronous master-slave single-ended serial bus. Thus I²C requires a clock signal, allows multiple master devices and multiple slave devices (master-slave) and the signal voltage is relative to the ground (single-ended). The I²C interface consists of two wires each for one of two signals: *synchronous clock* (SCL) for the communication synchronization; and *synchronous data* (SDA) for data transfer. I²C supports two communication speed: *standard* running at the 100 kHz; and *fast* at 400 kHz. The accelerometer ADXL345 provides only two possible I²C addresses based on the logical (high/low) value of the SDO/ALT ADDRESS signal (shown in Fig. 18). The SDO signal is required only by the SPI interface, hence the ALT ADDRESS can be used when the I²C interface is operational. The limitation of usable address forces us to develop a multiplexer printed circuit board described in the following section.

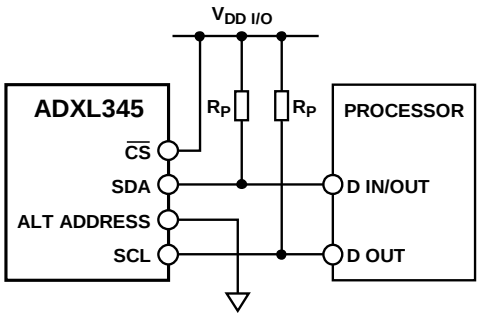


Figure 18: I²C connection diagram.[12]

Accessing the measured data is provided by six one-by registers[13](two bytes per each axis). There are two possible ways of reading from the registers using I²C: single-byte reading; and multiple-byte (batch) reading. The single-byte reading reads one byte on the given address, while n-byte reading sequentially reads n bytes starting on the given address ADD and ending for the address ADD+(n-1). Using six-byte reading is recommended to prevent changes in the stored data between each single-byte reading as it is described in [39].

[12]Modified version of Figure 8 from the ADXL datasheet [39] now respecting International Electrotechnical Commission standard

These six registers are used for accessing the measured data no matter which FIFO mode and measure mode is used. In the case of the bypass FIFO mode, the measured data are stored directly in these six registers. All other FIFO buffer modes send data from the top of the stack to these six data registers. Similarly to the bypass FIFO mode, using six-byte read is necessary to collect data from all theaxes, because reading any of six data register flushes all stored data in data registers and initiate loading another sample stored in FIFO. Note thatthe time delay of $5\ \mu s$ between readings from data registers is required for the communication speed higher than 1.6 MHz but as long as the maximum communication speed of $\mathrm{I^2C}$ is far below this value (maximum 400 kHz), this delay is not a concern for $\mathrm{I^2C}$.

4.5 Multiplexer Printed Circuit Board

There are two main reasons why multiplexer printed circuit board is required to be developed. Firstly, the Shifter Shield extension board provides only two $\mathrm{I^2C}$ communication interfaces, which is not enough for collecting data from all six ADXL345 accelerometers. This problem cannot be resolved by removing the Shifter Shield extension board, because Shifter Shield provides the power voltage level required by the accelerometers. Secondly, the used accelerometers do not provide the setting of their addresses, albeit it is possible to select an alternative address by connecting ALT_ADRESS to the power voltage instead of connecting it to the ground, as shown in Fig. 18. Using both $\mathrm{I^2C}$ interfaces provided by the Shifter Shield together with the combination of standard and alternative addressing would result in four possible-to-connect accelerometers, which is still insufficient for six accelerometers required for the hexapod walking robot.

Therefore, a pair of the Texas Instrument high speed CMOS multiplexers CD74HCT4051 for both $\mathrm{I^2C}$ signals (SDA and SCL) are utilized. Each of the multiplexer provides multiplexing up to 8 signals, but requires *multiplexing address* consisting of three address bits ($\mathrm{A_0}$, $\mathrm{A_1}$, and $\mathrm{A_2}$) and thus three general purpose input output (GPIO) pins of the Odroid address both multiplexers.

The required printed circuit board (PCB) has been designed in the Autodesk Eagle PCB design software. Connectors for the accelerometer wires and 6-pin header have been added to improve modularity of the PCB and enabling easy attachment of the connected accelerometers and replacement of malfunctioning accelerometer. The order of wires in the connector has been chosen to decrease a possible noise and crosstalk by placing both $\mathrm{I^2C}$ signals on one side of the connectors, the ground wires next to them, and the powered wires in the opposite side of the connector. The interrupt wire has been soldered into a one-pin connector, and it can be therefore attached regardless to other signal connectors. The interrupt signals are connected to six GPIO pins which are set up to be software interrupt pins.

Table 4: The order of the wires in the accelerometer connector.

	Wire name	Value of signal
1	Serial Clock (SCL)	variable
2	Serial Data (SDA)	variable
3	ALT ADDRESS	grounded
4	Ground (GND)	grounded
5	Supply Voltage	3.3 V ($\mathrm{V_{CC}}$)
6	Interface Supply Voltage	3.3 V ($\mathrm{V_{CC}}$)
I	INT1	variable

[13] DATAX0, DATAX1, DATAY0, DATAY1, DATAZ0 and DATAZ1

4.6 Central Inertial Unit

Inertial data measured by the accelerometers can be transformed (rotated) from the end-effector coordinates into the hexapod body coordinates, but even after this transformation based on the forward kinematics, the data remains relative to the rotation of the hexapod body. An inertial measurement unit attached to the body platform is required to define the rotation of body based on the gravity. The body orientation is necessary for obtaining the inertial measurements in the world coordinates. The deployed Xsens MTI-30 unit provides this capability and furthermore can serve as the Attitude Heading Reference System unit (AHRS) compensating drift from the gyroscopes integration by both gravity and the earth magnetic field to improve the precision of measurements.

Figure 19: Xsens MTI-30 inertial measurement unit.[14]

[14] Source: https://www.xsens.com/products/mti-10-series/

Chapter 5
Proposed Methods

In this chapter of the thesis, the proposed methods for the foot-strike detection using accelerometer measurements are presented. The proposed method is divided into five sections. First three sections correspond to data processing steps, the fourth section describes the data collection provided by the groundwork adaptive gait [2]. In the last section, aside from the foot-strike detection method, the proposed method for the terrain classification using the inertial data are be described.

The data processing pipeline consists of the following steps: the *data acquisition* followed by the *data processing*, and ended in the *foot-strike detection*. The data acquisition takes care of transferring data from the accelerometer to the central control unit. The data processing compensates differences in the leg position during the robot movement and the attachment of the accelerometer on the robot leg. The foot-strike detection determines whether leg hits the ground or not, and therefore, if leg should continue in movement or its motion should be stopped. The data acquisition and the detection methods, varies for the particular approach, while the data processing remains the same.

There are two different approaches presented in this thesis based on different measure modes of the accelerometer; the *stream* method (Fig. 20) is using the data from the continual reading of the acceleration measurements and the *single-tap* method (Fig. 21) is utilizing the data provided by the accelerometer ADXL345 with the namesake feature enabled. A similar terminology applies to classifiers (stream classifiers and single-tap classifiers) mode, inertial measurements (stream measure mode and single-tap measure mode) and the data (stream data and single-tap data). The differences between these proposed methods are the most obvious in the data acquisition step, therefore the characteristics of the stream method and the single-tap method are described in detail in the following section.

5.1 Data Acquisition

The data acquisition is the first step of data processing pipeline. The purpose of this step lies in the transferring of the measured data from the accelerometer to the central control unit necessary for the further processing.

In the ***stream mode*** (visualized in Fig. 20), the controller continuously reads the current data from the accelerometer. The main trait of the stream mode is setting the bypass FIFO mode, which disables the FIFO buffer and allows reading data instantly. Using a particular gait (such as the used tripod gait or tetrapod gait) results in the requirement for multiple reading done simultaneously as several legs can be in the swing phase at the same time. Therefore, the bandwidth of $\mathrm{I^2C}$ has been divided equally between the individual accelerometers. Three parameters need to be set in the accelerometer in the stream mode; the acceleration range, the resolution, and the data rate. The acceleration range $\pm 16g$ and the full resolution have been set to keep as much information provided by the accelerometer as possible. Setting 10-bit resolution would decrease the accuracy of the measurement, and smaller range would result in the information loss during the expected high peaks. The data rate of the stream mode is set to $200\ \mathrm{Hz}$ as the higher data rate is not possible because of the bandwidth of $\mathrm{I^2C}$. The standalone thread has been implemented for periodical data reading. Each $5\ \mathrm{ms}$, the current inertial data stored in the all accelerometers are read sequentially one by one while using multiplexing addresses (see Section 4.5). After collecting all the required data and passing them for further processing (lasting $\Delta t\ \mathrm{ms}$), the reading thread is put to sleep for the remaining time ($5 - \Delta t\ \mathrm{ms}$).

The ***single-tap mode*** (visualized in Fig. 21) depends on the interrupts generated by the ADXL345.

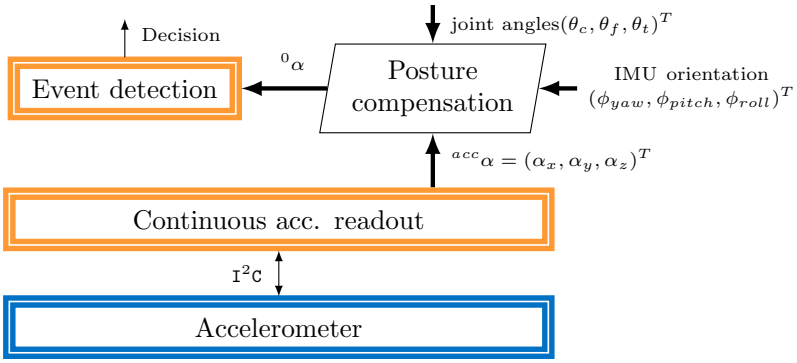


Figure 20: Foot-strike detection in the stream (continuous) operation mode.

The main trait of the single-tap mode is enabling the single-tap feature and enabling the trigger FIFO buffer mode (see Sections 4.4.1 and 4.4.2). The same measurement parameters as for the stream mode are required in the single-tap mode extended by those related to the single-tap mode and trigger FIFO mode. The acceleration range and the resolution remains the same as in the case of the stream mode, but the sampling rate was experimentally increased to 800 Hz. Two additional parameters are required for the single-tap measure mode; the minimum tap duration and the tap threshold for which the appropriate values have been found experimentally. The minimum tap duration is set to 6.9 ms, and the tap threshold is set to 4.9 g. The single-tap mode has been enabled and mapped to the interrupt signal `INT1`. Lastly, two parameters need to be set for the trigger FIFO mode; the triggering interrupt pin and the number of the samples before interrupt occurrence. The center of the single-tap peak has been set to be approximately in the center of the FIFO buffer by setting the value of 16 in the sample bits, and the triggering interrupt of the trigger FIFO mode has been set to the interrupt signal `INT1`. Unlike the stream mode, the single-tap mode requires a different approach in the data collection. All six GPIO (general purpose input output) pins chosen to be connected to the `INT1` interrupt signals are continually read in the endless cycle waiting for logic one to occur. Whenever the hardware interrupt appeared, all data stored in the FIFO buffer are read and send for processing followed by resetting the interrupt FIFO mode, which allows a new event to be classified.

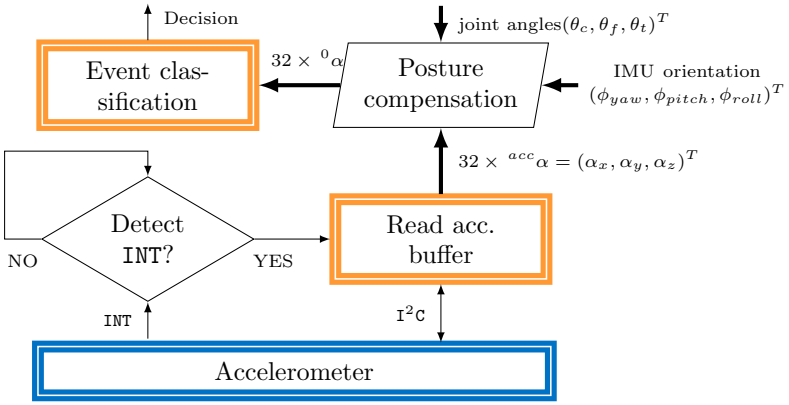


Figure 21: Foot-strike detection in the single-tap (interrupt) operation mode.

5.2 Data Preprocessing

Read accelerometer measurements have to be processed to unify the orientation of the measured data. Three particular reasons cause differences in the collected data. Firstly, in the rough terrains, the robot trunk orientation may vary based on several factors such as the slope of the traversed terrain. Secondly, the orientation of the accelerometer varies during the leg movement relative to the body. Lastly mounting of the accelerometer on the left and right legs is different.

Thus we are looking for the rotation matrix $\mathbf{R}$ that express the raw data ${}^{\text{acc}}\alpha = (\alpha_x, \alpha_y, \alpha_z)$ measured in the end-point coordinates of the global reference frame ${}^{0}\alpha$

$$ {}^{0}\alpha = \mathbf{R}\ {}^{\text{acc}}\alpha. \tag{6} $$

The nature of the particular reasons for different orientation of measured data allows splitting $\mathbf{R}$ into three rotation matrices corresponding to each rotation transformation

$$ \mathbf{R} = \mathbf{R}_{\text{body}}\mathbf{R}_{\text{leg}}\mathbf{R}_{\text{acc}}. \tag{7} $$

The joint angles $(\theta_c, \theta_f, \theta_t)$ of the respective leg along with the global orientation of the robot trunk $(\phi_{\text{yaw}}, \phi_{\text{pitch}}, \phi_{\text{roll}})$ and the constant coxa angle mount offset θ_c^{off} and the constant accelerometer mount angle β^{off} defines the rotation matrix $\mathbf{R}$. The body rotation based on the terrain slope is affected only by the rotation around x and y axes, therefore only ϕ_{pitch} and ϕ_{roll} change the accelerometer orientation resulting in (8). The joint angles $(\theta_{\text{c}}, \theta_{\text{f}}, \theta_{\text{t}})$ varying during the leg movement along with the constant coxa angle mount offset $\theta_{\text{c}}^{\text{off}}$ rotates the accelerometer data in the xy-plane and xz-plane as visualized in Fig. 13. These leg related rotations are expressed in (9). Lastly, the end-point coordinates are affected by the constant accelerometer mount angle β^{off} that differs for left and right legs, i.e., $0°$ and $180°$, respectively, as outlined in (10).

$$ \mathbf{R}_{\text{body}} = \mathbf{R}_y(\phi_{\text{pitch}})\mathbf{R}_x(\phi_{\text{roll}}), \tag{8} $$

$$ \mathbf{R}_{\text{leg}} = \mathbf{R}_z(\theta_{\text{c}} - \theta_{\text{c}}^{\text{off}})\mathbf{R}_y\left(-\theta_{\text{f}} - \theta_{\text{t}}\right), \tag{9} $$

$$ \mathbf{R}_{\text{acc}} = \mathbf{R}_y(\beta^{\text{off}}), \tag{10} $$

where $\mathbf{R}_x(\varphi)$, $\mathbf{R}_y(\varphi)$, and $\mathbf{R}_z(\varphi)$ are the rotation matrices around the respective axes by angle φ (11).

$$ \mathbf{R}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}, \quad \mathbf{R}_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix}, \quad \mathbf{R}_z(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11} $$

The global orientation $(\phi_{\text{yaw}}, \phi_{\text{pitch}}, \phi_{\text{roll}})$ of the robot is provided by the attitude heading reference system (AHRS) unit Xsens MTI30 attached to the hexapod trunk. The joint angles $(\theta_c, \theta_f, \theta_t)$ are set in the central control unit, and therefore accessible directly. The coxa angle mount offset $\theta_{\text{c}}^{\text{off}}$ and the accelerometer mount angle β^{off} do not vary during the robot movement; hence, they can be stored as constants specific to each leg. The transformed data are ready to be fed into the event detector in the case of the stream mode and into the event classifier in the case of the single-tap mode.

5.3 Foot-Strike Detection

After the preprocessing, data are unified and ready to be classified. Two classifiers are used for each operation mode; the *event detector* in the case of the stream operation mode and the *event classifier* in the case of the single-tap operation mode. This distinction between the event detector and event classifier is based on the fact that the stream operation mode produces a continuous time-equidistantly stream of data, therefore events (foot-strikes) are detected, whereas the single-tap operation mode provides discrete event data, which are not related in time, hence events are classified to foot-strikes.

Both classifiers may differ in classifier input data vector length. The data produced by the accelerometer in the single-tap operation mode always consists of 32 measurements resulting in a vector of size 96. This size of the single-tap operation mode vector originates in the size of the FIFO buffer. The i-th interrupt data obtained from the accelerometer in the single-tap operation mode is expressed as

$$\mathbf{v}_{\text{in}_i} = (\alpha_{i_{x_1}}, \alpha_{i_{y_1}}, \alpha_{i_{z_1}}, \alpha_{i_{x_2}}, \alpha_{i_{y_2}}, \alpha_{i_{z_2}}, \ldots, \alpha_{i_{x_{32}}}, \alpha_{i_{y_{32}}}, \alpha_{i_{z_{32}}}). \tag{12}$$

The stream operation mode allows only reading one sample at the time, which does not provide enough information for the event detector, therefore the *sliding window* containing more than one sample is required. The *sliding window* in the current context means a buffer storing the last n samples. Each time the new data are collected, the content of the sliding window is updated; the new data are added on the beginning, the older data slides to right and the oldest sample are removed from the view. There is no direct restriction on the size of the sliding window, but we stick to the same number of samples as in the case of the single-tap operation mode. The input vector of the event detector in the time t with the sample rate τ is shown in (13).

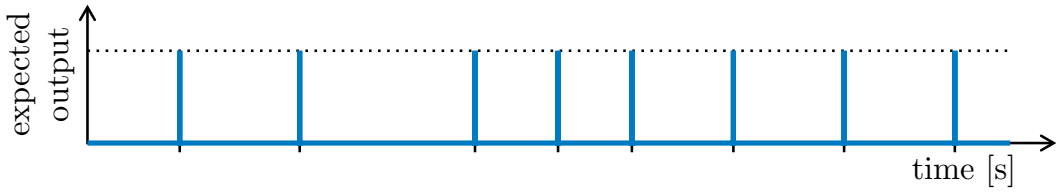
$$\mathbf{v}_{\text{in}}[t] = (\alpha_x[t], \alpha_y[t], \alpha_z[t], \alpha_x[t-\tau], \alpha_y[t-\tau], \alpha_z[t-\tau] \ldots, \alpha_y[t-31\tau], \alpha_z[t-31\tau]) \tag{13}$$

The input vectors for both operation modes were then fed into event detector and event classifier. We considered three classification methods in the total; Support Vector Machine (SVM) for both operation modes, Long-Short Term Memory (LSTM) recurrent neural network learning using the Back Propagation Through Time (BPTT) for the stream operation mode and finally a simple multilayer Neural Network (NN) for event classification of the single-tap operation mode.

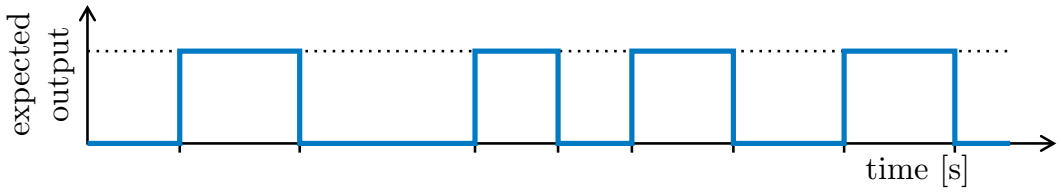
5.4 Data Collecting and Learning

The most important part of the foot-strike detection is the learning of the detector/classifier using real data. The learning data was collected while the hexapod was guided through the area made from wooden blocks of variable height. Both data from the accelerometers and the foot-strikes events detected by the adaptive gait [2] were recorded while the robot traverses the rough terrain. The data collected from the groundwork was used as the reference and helps us to label the collected accelerometer data. We collected 1332 events in the stream operation mode and 1401 events in the single-tap mode in total. The recorded data were then split into the training and testing data in the ratio 0.7:0.3.

Several issues arose during the learning process for both measure modes. Firstly, data collected in the stream measure mode are unbalanced (one foot-strike event per several hundreds of samples). Hence we decided to modify the raw data (Fig. 22a) to alternate between zero and one whenever a foot-strike event occurred. The modified data (Fig. 22b) are then used for the Neural Network event detector. The modified data (Fig. 22b) made possible to use standard learning approaches with the root mean squared standard minimization as the objective function for the NN event detector. Secondly, differences in the sampling rate of the relatively slow groundwork adaptive gait and the higher sampling rate of accelerometers in the stream measure mode resulted in the fact that there are several accelerometer measurements samples between two adaptive gait foot-strike classification labels. Therefore we have decided to expand the peak in the raw data to the k-nearest accelerometer samples, where k is determined based on the number of accelerometer measurements between two adaptive gait samples. The labeled data with the expanded label peaks is used for learning the SVM classifier using a sliding window. Lastly, using the interrupt based asynchronous data collected in the single-tap measure mode requires assigning each of the event detected by the accelerometer to the appropriate label. Hence, each single-tap data are labeled based on the closest adaptive gait data available. This method of labeling the single-tap data is used for the both SVM and feed-forward NN event classifiers.



(a) Unbalanced raw foot-strike data in the stream measure mode.



(b) Modified foot-strike data in the stream measure mode.

Figure 22: Mean value of the raw foot-strike data (a) is close to zero, therefore the loss used in the BP is small in each learning epoch, hence the learning process is slowed down. On the other hand the using alternating data (b) with the mean value close to 0.5 increase the loss used in the BP and improve the learning efficiency.

5.5 Terrain classification method

The terrain classification from to inertial data has been tackled by S. Otte et al. in [19] for the wheeled robot and by F. L. G. Bermudez et al. in [10] for the legged robots. We believe the most information about the terrain are contained in the inertial data at the moment of the foot-strike, therefore the single-tap measure mode has been used for collecting data used in the terrain classification task. The settings of the single-tap is identical to the setting for the single-tap based foot-strike detection and the collected raw data have been unified as described in Section 5.2. The unified data are then fed into the feed-forward neural network with a single hidden layer. The size of the input layer of the proposed NN is same as the number of the samples provided by the accelerometer ADXL345 and each classified terrain class has the corresponding output neuron.

Chapter 6
Results

The proposed ground detection methods have been experimentally evaluated in two scenarios. Both scenarios used real data collected by the hexapod walking robot with the groundwork adaptive gait [2]. The performance of the individual detection methods examined in the first scenario have been evaluated on the datasets collected using the adaptive motion gait. The best performing ground detection method from the first scenario has been then deployed on the hexapod platform in the second scenario. The performance of our approach has been tested in several experimental trials and then compared to the performance of the groundwork [2]. In addition to the foot-strike detection, the possibility of using inertial data for the terrain classification is examined in the third scenario.

6.1 Learning scenario

The performance of the individual methods has been statistically evaluated on the data collected on rough terrain laboratory test-track shown in Fig. 23 using the groundwork method [2]. In our laboratory, the test-track is the widely used way to approximate expected real-life rough terrains. The whole terrain test-track of the dimensions 2.5×1.2 m consists of the wooden frame and over 200 wooden 10×10 cm blocks that vary in slope and height. The accelerometer data along with ground-detection provided by adaptive motion gait [2] has been collected while the robot has been guided by an operator across the experimental test-track. The accelerometer data has been then used as the input of the classification methods, where the ground-detection presents the expected output of the classification methods. The different settings of the measure modes required recording datasets for the single-tap measure mode and the stream mode separately, hence the stream dataset has been collected with 1332 foot-strike events followed by the single-tap dataset consisting of 1401 foot-strike events. These datasets have been used for individual learning method and the experimental tuning of the parameters of the methods as described in Section 5.4.

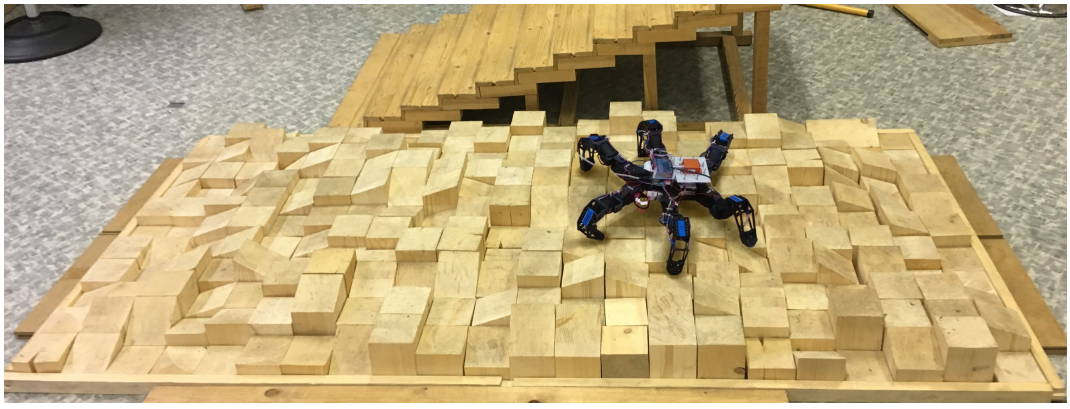


Figure 23: Hexapod in the test-track used to simulate real-life scenarios.

For the ***stream*** measure mode, the SVM classifier and the LSTM NN classifiers have been used. The SVM classifier with 32 elements wide sliding window for the accelerometer data ${}^0\vec{\alpha}$ has been learned to detect the foot-strike events. Multiple kernel functions have been tested: the polynomial kernel function with the degree in the range $\langle 3, 10 \rangle$ and the radial basis function (RBF). Later, the RBF has been chosen to be used as the kernel function because the polynomial kernel functions be-

Table 5: Detection results

	Stream		Single-tap	
	SVM	LSTM	SVM	NN
Precision	0.63	0.57	0.81	0.78
Recall	0.31	0.15	0.85	0.65

have poorly on the given data in comparison to the RBF. Next, the Long-Short Term Memory neural network (LSTM NN) has been trained using the Backpropagation Through Time (BPTT). The LSTM consists of 3 inputs ($\alpha_x, \alpha_y, \alpha_z$), 32 hidden states and a single output. The sigmoid function

$$\sigma(x) = \frac{1}{1 - e^{-x}} \tag{14}$$

is used as the activation function. The data modified as described in Section 5.4 and fed into the LSTM give the expected output.

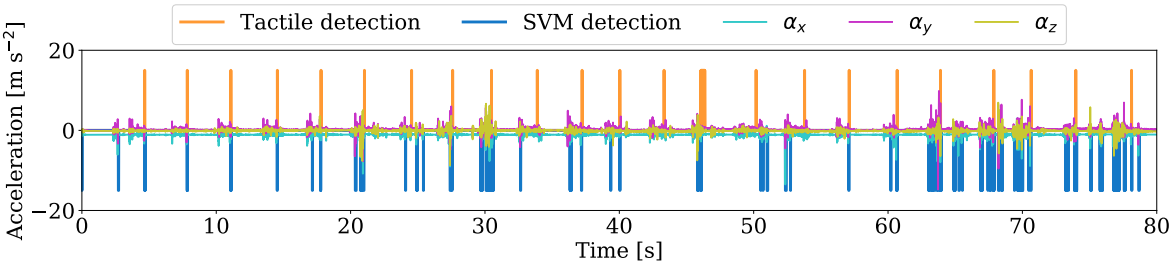
For the ***single-tap*** measure mode, the SVM classifier has been used together with the feed-forward Neural Network (NN). Similarly to the stream measure mode, the SVM uses the radial basis function (RBF) as the kernel function and size of the input data ${}^0\vec{\alpha}$ is 32 elements as well, but different nature of the collected data discussed in Section 5.3 requires individual classifier for the single-tap measure mode. Along with the SVM classifier, the feed-forward Neural Network (NN) has been trained using the backpropagation (BP). The feed-forward Neural Network consists of 32 inputs, single hidden layer of the same size and the single output using the sigmoid activation function $\sigma(x)$ (14).

The classification results of the proposed classifiers are summed up in Tab. 5 and visualized in Fig. 24. Classifiers based on the single-tap measure mode have been more successful than the classifiers using the stream measure mode for the reasons discussed further in this section. The single-tap SVM classifier (Fig. 24b) has reached the best performance and has been used in the following scenario. The feed-forward NN using the single-tap (Fig. 24c) data provides the competitive results to the single-tap SVM, unlike the LSTM neural network in the stream mode, which provides poor results even on the modified data. The stream measure mode based SVM classifier (Fig. 24a) have been more precise than the LSTM neural network, even though it has not been as successful as the single-tap based classifiers.

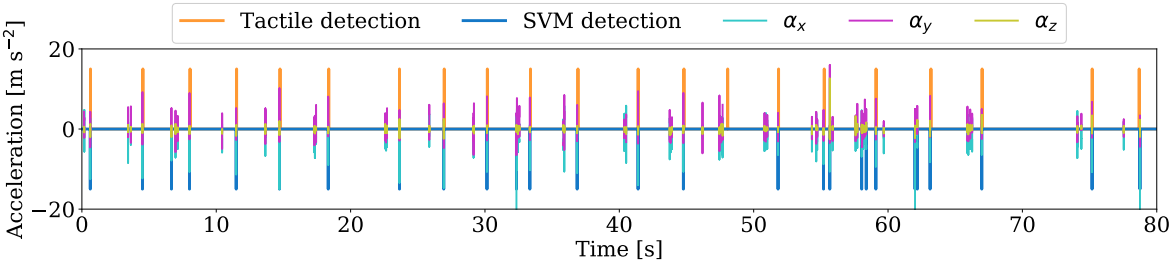
The possible explanation of the different success rates of the stream measure mode based classifiers and the single-tap measure mode based classifiers lies in the different sampling rate of the stream measure mode (200 Hz) and the single-tap measure mode (800 Hz). The foot-strike event can be simply missed in the stream mode because of the lower sampling frequency, which cannot be further increased because of the already reached limit of the $\mathrm{I}^2\mathrm{C}$ bandwidth. On the other hand, in the single tap measure mode, only the relevant data are transferred reducing the $\mathrm{I}^2\mathrm{C}$ usage and allowing the higher measure rate. Let's further note that the single-tap method provides a fusion between rule-based detection algorithms [12] and algorithms based on machine learning [22]. Hence, the basic rule-based decision algorithm implemented right in the chip (see Section 4.4.1) roughly filters out some of the events and pass relevant data into the advanced machine learning classifier. This two-step classification might also improve the overall performance of the single-tap approach.

6.2 Testing scenario

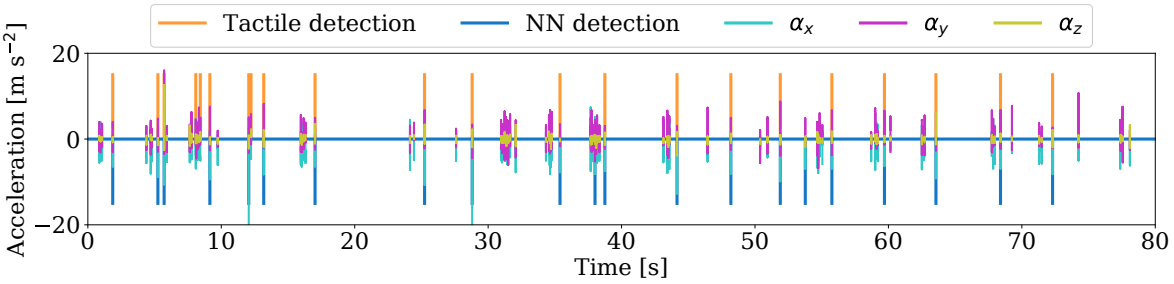
Regarding the results in the previous section, the single-tap SVM classifier is chosen to be deployed into the hexapod central control unit and further evaluated in the additional scenario. The alternative inertial adaptive gait is implemented by replacing the force threshold-inspired position control [2] in the adaptive gait by our proposed classifier. Our inertial adaptive gait uses the feedback solely from



(a) The SVM classification results using the data from the stream measure mode.



(b) The SVM classification results using the data from the single-tap measure mode.



(c) The NN classification results using the data from the single-tap measure mode.

Figure 24: The comparison of the classification method; the SVM using data measured in the stream measure mode (a), the SVM using the data obtained in the single-tap measure mode (b) and the NN using the data measured in the single-tap mode (c).

the accelerometer without any additional sensors nor measurements including servo position used by the FTP-inspired approach.

In this scenario, the performance of the proposed classifier has been examined in two experiments, in which the robot has been requested to traverse both flat terrain and the rough terrain in the test-track using only new implemented inertial adaptive gait. Firstly, the general test of the proposed classifier has been carried out. The robot was placed on the flat surface and its ability of movement was observed. As expected, the performance of the hexapod was sufficient enough to traverse the flat terrain, so the hexapod was moved to the test-track (Fig. 23) to approach the rough terrain. The proposed classifier was able to detect foot-strikes in the rough terrain and the robot was able to traverse rough terrain, which support the expected benefits of using inertial measurements in the foot-strike detection over the FTP-inspired detection. Beside the verification, we focused on the further improvement of the proposed classifier, and therefore, we started to gradually increase the speed of the motion aiming to reach the limits of the classifier and examine the generalization level of the classifier. Increasing the motion speed requires certain level of the generalization of the learned classifier because changing speed of the leg movement during the swing down phase results in slightly different accelerometer data. Before proceeding to the details of the speeding up process, let's recall that without any feedback, the robot would be unable to traverse any form of rough terrains. The speeding up process

was split into two stages; speed increasing and performance measurements. The speed of the forward movement has been in each iteration of the speeding cycle slightly increased or decreased using the binary-like search using results from the previous performance (e.g., increase in the case of improvements et vice versa). After the finding the optimal speed, the two experiments have been carried out.

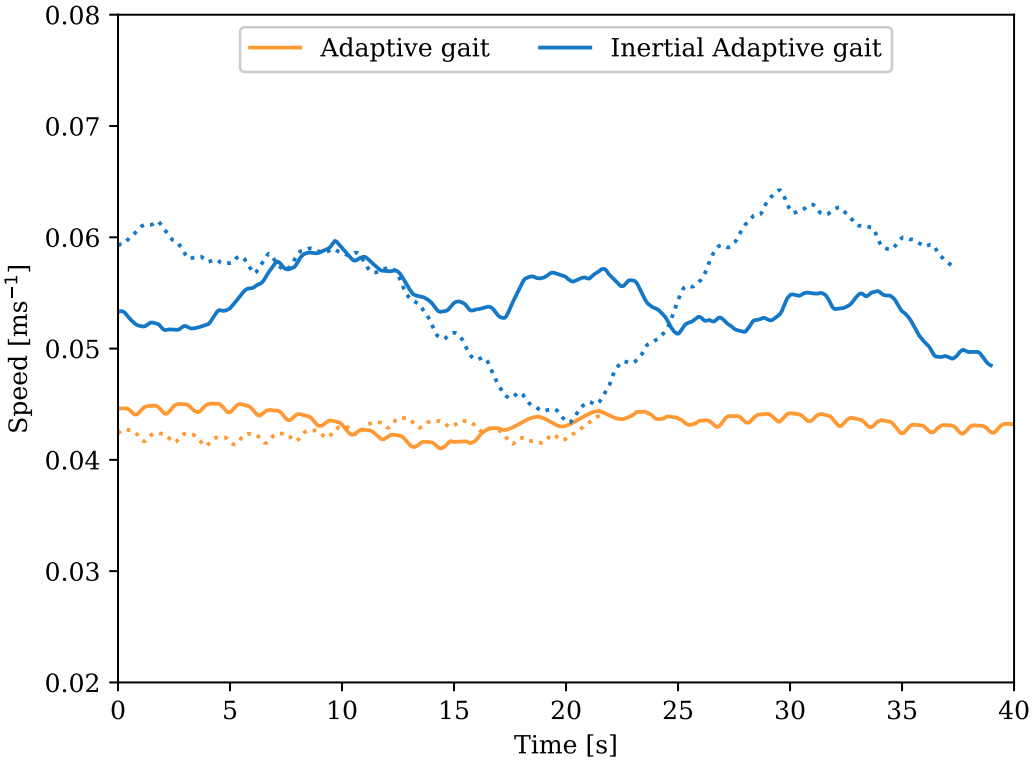
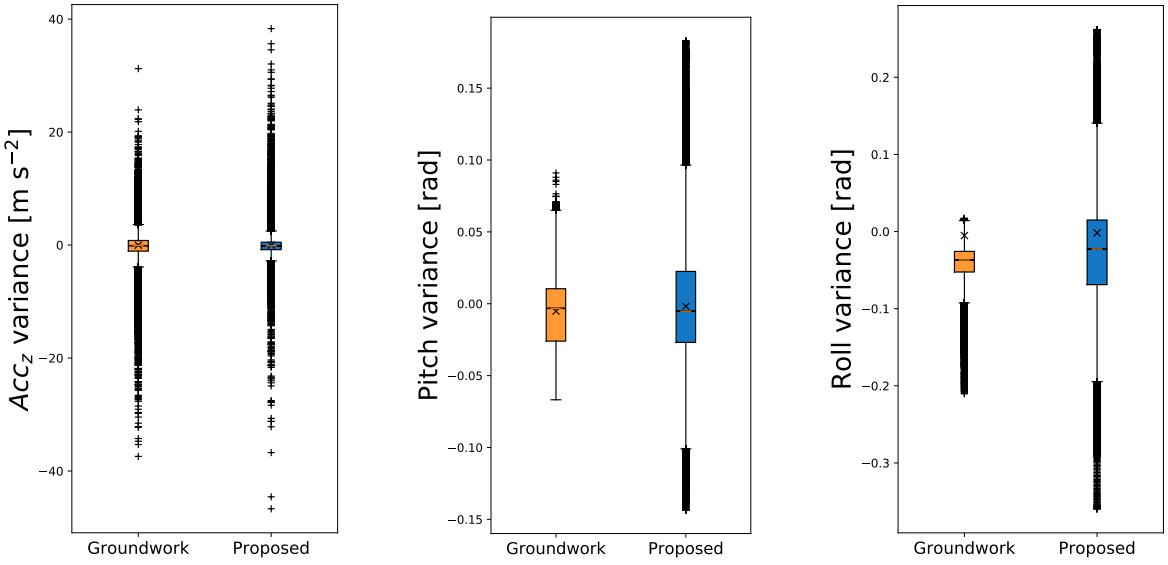


Figure 25: An illustrative comparison of the hexapod speed for the groundwork adaptive gait (orange) and the proposed inertial adaptive gait (blue).

In the first experiment, the speed and the stability of the proposed classifier have been compared to the groundwork. Two types of data has been collected; the inertial and the visual. The visual data have been used for the position localization and speed calculation for both classifiers and the inertial data have been used for the stability comparison. The robot has performed ten guided trials of traversing flat terrain using inertial adaptive gait and ten guided trails using the adaptive gait [2].

The position of the robot was tracked by the ceiling-mounted visual fiducial system [40] using the April tag added to the top of the hexapod. The visual data has been then processed, the trajectory has been reconstructed and the speed has been calculated for both adaptive gait and the proposed inertial adaptive gait. The changes of the speed over time are visualized in Fig. 25 for the hexapod using the groundwork adaptive gait and the speed of the hexapod using our inertial adaptive gait as well. The average speed and the standard deviation have been calculated as follows. The groundwork adaptive gait [2] $v_{\mathrm{g}} = (0.038 \pm 0.006)\ \mathrm{ms}^{-1}$ and the proposed inertial adaptive gait $v_{\mathrm{p}} = (0.056 \pm 0.007)\ \mathrm{ms}^{-1}$. On the flat surface, usage of the inertial adaptive gait provides 1.47 times locomotion speed up of the speed of the groundwork adaptive gait.

The stability of the robot during the locomotion is greatly influenced by the reliability of the ground detection. Therefore we have experimentally verified the reliability of the proposed method by observing the body motion. The additional inertial measurement unit Xsens MTi-30 has been mounted to the hexapod body to measure the body orientation and acceleration. The inertial data



(a) Acc_z variance (b) Pitch variance (c) Roll variance

Figure 26: The five-number summary of the performance indicators: the linear acceleration in the z-axis (a), the pitch (b) and the roll (c) angles for the groundwork FTP-inspired classifier (orange) and the proposed classifier (blue).

has been collected with the 400 Hz during the robot locomotion. Then, the linear acceleration in the z-axis and the pitch and the roll angles have been chosen as the performance indicators. The variances of the performance indicators have been computed. The five-step summary for the linear acceleration in the z-axis and the pitch and the roll angles are visualized in Fig. 26. The average linear acceleration in the z-axis is similar for the both classifiers and the standard deviation is surprisingly higher for the groundwork classifier than for the proposed classifier: the groundwork adaptive gait $a_{z_{\mathrm{g}}} = (-0.22 \pm 2.76)$ ms^{-2} and the proposed inertial adaptive gait $a_{z_{\mathrm{p}}} = (-0.12 \pm 1.73)$ ms^{-2}. For both measured angles, the standard deviation is higher for the proposed method as expected from the visual observation of the uneven motion of the hexapod using the proposed classifier. The pitch and the roll for the groundwork methods are $\phi_{\mathrm{roll_g}} = (-0.52 \pm 2.50) \times 10^{-2}$ rad and $\phi_{\mathrm{pitch_g}} = (-4.32 \pm 2.86) \times 10^{-2}$ rad, respectively. The pitch and the roll for the proposed classifier are $\phi_{\mathrm{roll_p}} = (-0.18 \pm 4.04) \times 10^{-2}$ rad and $\phi_{\mathrm{pitch_p}} = (-2.87 \pm 7.71) \times 10^{-2}$ rad respectively.

In the last step, the performance in the rough terrain using the test-track was examined. The robot has performed five guided trials of traversing the laboratory test-track for the adaptive gait and the proposed inertial adaptive gait, while traverse time was measured and used to quantify the progress. We have experimentally found that reached data generalization allows to speed up the robot locomotion up to 1.7 times the maximum speed of the FTP-based method used in the groundwork [2] in the rough terrains. The absolute value of the overall forward speed was increased from 0.03 ms^{-1} for the groundwork approach up to 0.05 ms^{-1} for our approach.

6.3 Terrain classification scenario

In the last scenario, the possible usage of the inertial measurements for the terrain classification have been examined. The scenario was divided into two parts. Firstly, the data from various terrains have been collected, classifier has been created and learned. Secondly the classifier has been tested.

In the first part of this scenario, the robot has been guided through four types of the terrains: flat floor; soft flat terrain represented by the artificial turf; the rough terrain represented by the test-

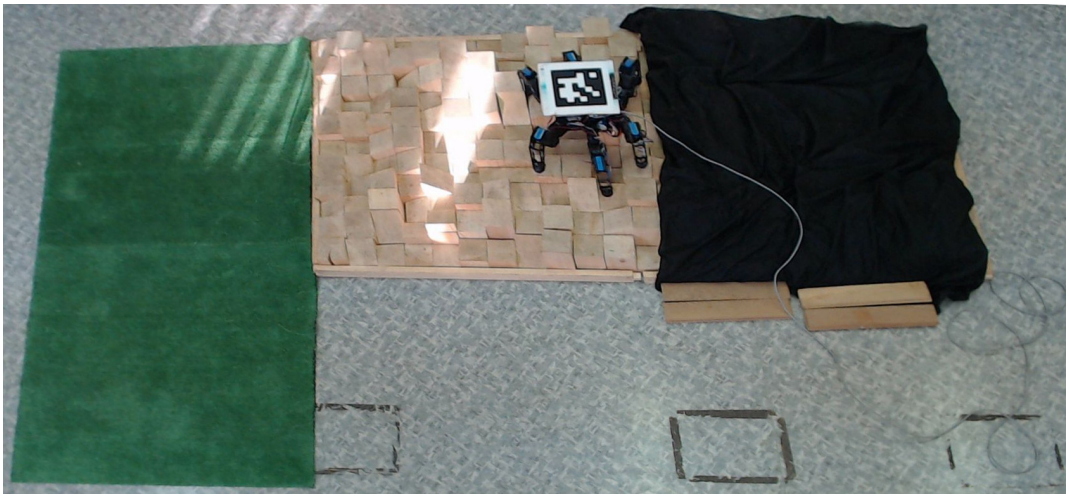


Figure 27: The terrain classification scenario overview. Four types of terrains are present: flat (floor), soft left (artificial turf, on the left), rough (test-track, upper center) and the rough soft (test-track covered by the black cloth, upper right). The robot is tracked using the April tag attached to the top of the hexapod.

track; and rough soft terrain represented by the test-tracked covered by the black fabric. All types of the terrain are visualized in Fig. 27. For each terrain, separate data set has been collected using the groundwork adaptive gait. In total, 5694 unified single-tap events have been collected for all four terrain classes. The data have been split into the training data and the testing data in the ratio 0.7:0.3 and labeled in hot-one encoding based on the terrain and the neural network has been learned.

Table 6: Confusion matrix

	Flat	Flat Soft	Rough	Rough Soft
Flat	0.91	0.06	0.03	0
Flat Soft	0.02	0.91	0.07	0
Rough	0.05	0.20	0.75	0
Rough Soft	0.08	0.33	0.59	0

The first part of the terrain classification scenario provides unexpected results summed in the confusion matrix depicted in Tab. 6. Even after experimenting with the combination of the multiple parameters such as the number of the hidden layers, the loss function used in the learning process and the used matrix the neural network was unable to classify the soft rough terrain (test-track covered by the black fabric). In two cases, changing the parameters resulted in the absolutely wrong classification of the flat soft terrain (flat artificial turf), while the soft rough terrain was classified relatively well. These results leads us to thought, that the similarities in the inertial data obtained in the soft flat terrain and the soft rough terrain are beyond the distinguishing abilities of the neural network and the difference in the parameter affect the preferred class. On the other hand, the neural network was able to distinguished between the flat terrain (floor) and the soft flat terrain (flat artificial turf).

In the second step of the terrain classification scenario, the previously developed terrain classifier has been tested in the real-life-like scenario. The robot robot has been guided through the area (Fig. 27) covered by four previously mentioned terrain. The position of the robot has been tracked by the ceiling-mounted visual fiducial system [40] using the April tag added to the top of the hexapod, similarly to the testing of the foot-strike classifier. The data collected using the inertial measurements

have been then classified using the terrain classifier. The classified data have been processed using the majority of three and then the classification results have been paired with the points obtained from the tracker. The points with the classified terrain and the original terrain are visualized in Fig. 28. The rough soft terrain has not been classified at all, as expected from the previous step of this scenario. The flat soft terrain has been classified most successfully from all chosen terrains and the rough soft terrain has been mostly classified as the flat soft terrain. Surprisingly, the classification of the flat terrain has not been as successful as predicted by the previous results.

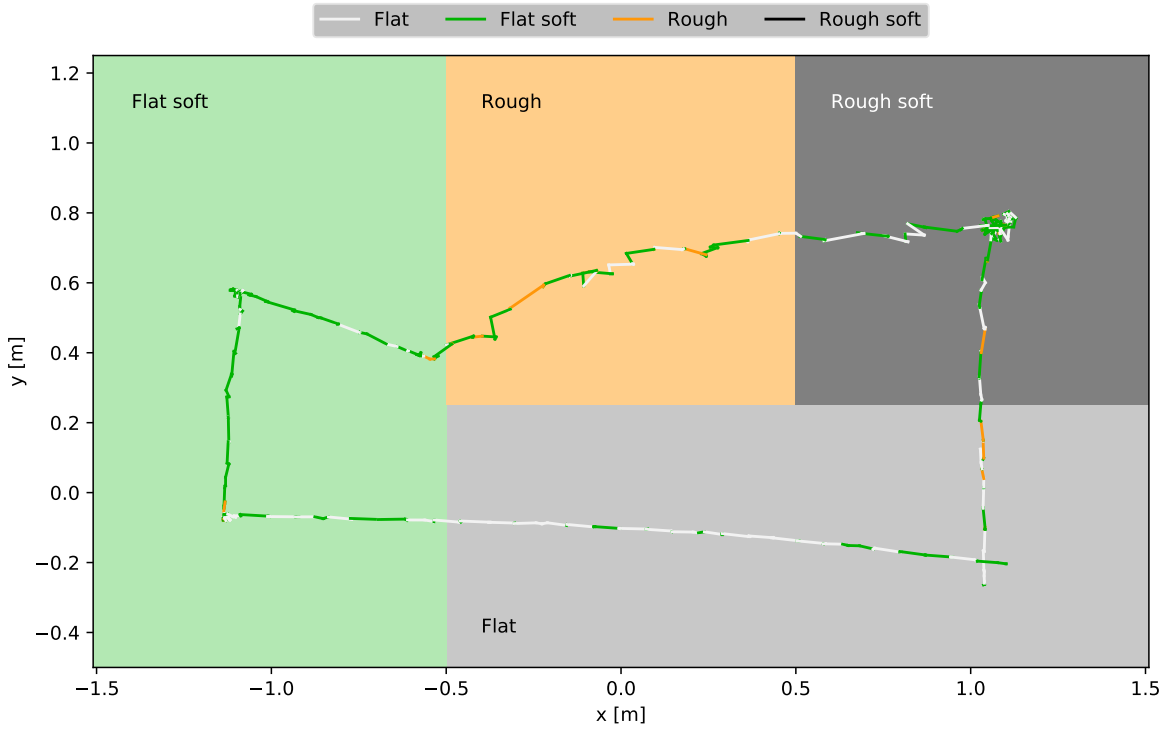


Figure 28: The visualization of the hexapod trajectory over the different terrains: flat (gray), flat soft (green), rough (orange), and rough soft (dark gray). The color of the trajectory corresponds to the classified terrain as labeled in the legend. The rough soft terrain has not been detected at all.

Chapter 7
Conclusion

In this work, possible advantages of using solely inertial measurements for the foot-strike detection of the hexapod walking robot in rough terrain has been examined and then compared with the method proposed in the groundwork [2] based on adaptive motion gait which uses only feedback from the servomotors. Beside commonly used event detection using the stream of accelerometer data, the event classification based on the *single-tap* feature provided by the ADXL345 accelerometers has been investigated and experimentally evaluated. In the total, four classifiers have been developed; the Long-Short Term Memory event detector for stream data, feed-forward Neural Network event classifier for single-tap interrupt based data and SVM for both types of data. The classifiers based on the single-tap data provided better results in comparison to stream data classifiers, therefore regarding the presented results, it seems that the usage of rule-based filtration of inertial data inside the accelerometer itself is more suitable and may improve the overall performance of the classifiers. The Support Vector Machine classifier for single-tap (the best performing classifier) was deployed on the hexapod walking robot and then tested in scenarios similar to the real-life traversing rough terrains. The proposed foot-strike detection method is not only able to achieve the same results as groundwork but also surpasses its ancestor and allows to increase the hexapod locomotion speed. The increasing of the locomotion speed led to 1.5 speedup of the overall forward speed in the flat terrains and up to 1.7 speedup in the rough terrain traversing. Further improvements in learning methods are required to improve the performance and additional speed up the locomotion in difficult terrain, hence we suggest using self-supervised learning of the event classifier based on the evaluation of the locomotion stability. Beside the foot-strike detection, the possible usage of the inertial measurements for the terrain classification have been examined. The terrain classifier using the neural network has been developed and tested on four types of the terrain. Even though the classifier has been able to distinguish between certain terrains (flat and flat soft), the further improvements and possibly adding data measured by different devices are required to improve the classification capabilities of the terrain classifier.

The hexapod walking platform is presented in Section 3.4 and the locomotion of the robot is described in the first part of the problem statement. The methods of the foot-strike detection and the terrain classification using the inertial measurements are introduced in Section 2.1 and the inertial data processing is described in Section 5.2. Three suitable classification methods (SVM, NN and LSTM NN) are chosen for the foot-strike detection and are described in detail in Section 4.6 and their results are reported in the first part of Section 5.5. The best performing classifier are compared to the groundwork method in the second part of Section 5.5. The possible usage of the terrain classification is outlined at the end of Section 4.6 and the terrain classification results are presented in the last part of Section 5.5. All given tasks of this thesis have been successfully accomplished.

References

[1] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *International Journal of Robotics Research (IJRR)*, vol. 20, no. 7, pp. 616–631, 2001.

[2] J. Mrva and J. Faigl, "Tactile sensing with servo drives feedback only for blind hexapod walking robot," in *10th International Workshop on Robot Motion and Control*, 2015, pp. 240–245.

[3] M. R. Fielding and G. R. Dunlop, "Omnidirectional hexapod walking and efficient gaits using restrictedness," *The International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 1105–1110, 2004.

[4] B. Huang, M. Chen, X. Shi, and Y. Xu, "Gait event detection with intelligent shoes," in *International Conference on Information Acquisition*, 2007, pp. 579–584.

[5] I. P. I. Pappas, M. R. Popovic, T. Keller, V. Dietz, and M. Morari, "A reliable gait phase detection system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, no. 2, pp. 113–125, 2001.

[6] I. P. I. Pappas, T. Keller, S. Mangold, M. R. Popovic, V. Dietz, and M. Morari, "A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole," *IEEE Sensors Journal*, vol. 4, no. 2, pp. 268–274, 2004.

[7] A. Miller, "Gait event detection using a multilayer neural network," *Gait & Posture*, vol. 29, no. 4, pp. 542–545, 2009.

[8] K. Walas, D. Kanoulas, and P. Kryczka, "Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 133–140.

[9] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto, "Haptic sensing foot system for humanoid robot and ground recognition with one-leg balance," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, pp. 3174–3186, 2011.

[10] F. L. G. Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing, "Performance analysis and terrain classification for a legged robot over rough terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 513–519.

[11] L. Vodovnik, A. Kralj, U. Stanic, R. Acimovic, and N. Gros, "Recent applications of functional electrical stimulation to stroke patients in ljubljana," *Clinical orthopaedics and related research*, vol. 131, p. 64—70, 1978.

[12] J. K. Lee and E. J. Park, "Quasi real-time gait event detection using shank-attached gyroscopes," *Medical & Biological Engineering & Computing*, vol. 49, no. 6, pp. 707–712, 2011.

[13] N. Mijailovi, M. Gavrilovi, and S. Rafajlovi, "Gait Phases Recognition from Accelerations and Ground Reaction Forces : Application of Neural Networks," vol. 1, no. 1, 2009, pp. 34–36.

[14] M. Djuric, "Automatic recognition of gait phases from accelerations of leg segments," in *9th Symposium on Neural Network Applications in Electrical Engineering*, 2008, pp. 121–124.

[15] D. J. van den Heever, K. Schreve, and C. Scheffer, "Tactile sensing using force sensing resistors and a super-resolution algorithm," *IEEE Sensors Journal*, vol. 9, no. 1, pp. 29–35, 2009.

[16] R. Williamson and B. J. Andrews, "Gait event detection for fes using accelerometers and supervised machine learning," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 3, pp. 312–319, 2000.

[17] H.-Y. Lau, K.-Y. Tong, and H. Zhu, "Support vector machine for classification of walking conditions using miniature kinematic sensors," *Medical & Biological Engineering & Computing*, vol. 46, no. 6, pp. 563–573, 2008.

[18] S. Došen and D. B. Popović, "Accelerometers and force sensing resistors for optimal control of walking of a hemiplegic," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1973–1984, 2008.

[19] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 5603–5608.

[20] J. Han, H. S. Jeon, B. S. Jeon, and K. S. P. G, "Gait detection from three dimensional acceleration signals of ankles for the patients with parkinson ' s disease," vol. 2628, 2011, pp. 1–4.

[21] A. T. M. Willemsen, F. Bloemhof, and H. B. K. Boom, "Automatic stance-swing phase detection from accelerometer data for peroneal nerve stimulation," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 12, pp. 1201–1208, 1990.

[22] Y. Shimada, S. Ando, T. Matsunaga, A. Misawa, T. Aizawa, T. Shirahata, and E. Itoi, "Clinical application of acceleration sensor to detect the swing phase of stroke gait in functional electrical stimulation," *The Tohoku Journal of Experimental Medicine*, vol. 207, no. 3, pp. 197–202, 2005.

[23] V. N. Vapnik and H. Siegelmann, "Support vector networks," 1963.

[24] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[25] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[26] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug 2004.

[27] Scikit-learn developers. (2017) Support vector machines. Sci-kit learn. Accessed on 2018-05-16. [Online]. Available: http://scikit-learn.org/stable/modules/svm.html#support-vector-machines

[28] L. I. R. Mark A. Aizerman, Emmanuel M. Braverman, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[29] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[30] M. Nielsen, *Neural Networks and Deep Learning*, 2015, accessed on 2018-05-16. [Online]. Available: http://neuralnetworksanddeeplearning.com/

[31] C. Olah. (2015) Understanding lstm networks. Accessed on 2018-05-18. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] D. O. Hebb, *Organization of Behavior*. Wiley & Sons., New York, 1949.

[35] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard University, 1975.

[36] I. Sutskever, "Training recurrent neural networks," Ph.D. dissertation, University of Toronto, 2013.

[37] *Dynamixel AX-12 user manual*, ROBOTIS, 2006, rev. 0, accessed on 2018-04-23. [Online]. Available: http://www.trossenrobotics.com/images/productdownloads/AX-12(English).pdf

[38] *User manual Odorid XU-4*, Hard Kernel, 2015, rev. 20170310, accessed on 2018-04-23. [Online]. Available: https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf

[39] *Digital Accelerometer ADXL345*, Analog Devices, 2009, rev. E, accessed on 2018-04-23. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf

[40] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *nternational Conference on Robotics and Automation (ICRA)*, 2011, pp. 3400–3407.