

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL AVIATION UNIVERSITY

Faculty of Aeronavigation, Electronics and Telecommunications

Department of computer integrated complexes

ADMIT TO DEFENSE

Head of the graduating department

_____ Viktor M. Sineglazov

“ _____ ” _____ 2024y.

QUALIFICATION WORK

(EXPLANATORY NOTE)

OF THE GRADUATE OF THE EDUCATIONAL DEGREE

“BACHELOR”

Specialty 151 "Automation and computer-integrated technologies"

Educational and professional program "Computer-integrated technological processes and production "

Theme: Quadruped robot walking algorithm

Performer: student of FAET-402 group Innokov Oleksandr Olegovich

Supervisor: Candidate of Technical Sciences, Tupitsin Mykola Fedorovych.

Norm controller: _____ Filyashkin M.K.

(sign)

Kyiv – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно-інтегрованих систем

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

_____ Віктор СИНЕГЛАЗОВ

“ ____ ” _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”

Спеціальність 151 "Автоматизація, та комп'ютерно-інтегровані технології"

Освітньо-професійна програма "Комп'ютерно-інтегровані технологічні процеси і виробництва"

Тема: Алгоритм ходьби чотириноного робота

Виконавець: студент групи КП-404Ба Інноків Олександр Олегович

Керівник: Кандидат технічних наук, Тупіцин Микола Федорович

Нормоконтролер: _____ Філяшкін М.К.

(підпис)

NATIONAL AVIATION UNIVERSITY

Faculty of Aeronautics, Electronics and Telecommunications

Department of aviation computer-integrated systems

Educational degree: Bachelor

Specialty 151 "Automation, computer-integrated technologies and robotics"

Educational and professional program "Computer-integrated technological processes and production"

APPROVED

Head of department

_____ Sineglazov V.M

“ _____ ” _____ 2023.

TASK

For the student's thesis

by: Innokov Oleksandr Olegovich

- 1. Thesis topic** (project topic) «Quadruped robot walking algorithm»
- 2. Deadline for an execution of a project:** from April 1 of 2024 to June 20 of 2024
- 3. Initial data for the project:** S. Talebi, M. Buehler and E. Papadopoulos. Towards Dynamic Step Climbing For A Quadruped Robot with Compliant Legs//Conference Paper in Proceedings - IEEE International Conference on Robotics and Automation · May 2013 DOI: 10.1109/ICRA.2013.6631035
- 4. Contents for explanatory note:**1) Review and analysis of movement algorithms for mobile robots;2) Description of the movement features of walking robots;3) Problem statement;4) Kinematic model of quadruped robot movement;5) Development of a robot movement algorithm;6)

Example of calculating the movement of a quadruped robot in a straight line;7) Analysis of the obtained results.

5. List of required graphic material: Figures, tables and diagrams.

6. Calendar schedule-plan:

№	Task	Execution term	Execution mark
1.	Getting the task	01.04.2024 – 04.04.2024	Done
2.	Formation of the purpose and main objectives of the study	05.04.2024 – 24.04.2024	Done
3.	Review and analysis of movement algorithms for mobile robots.	25.04.2024 – 01.05.2024	Done
4.	Description of the movement features of walking robots.	02.05.2024 – 10.05.2024	Done
5.	Problem statement;	11.05.2024 – 25.05.2024	Done
6.	Kinematic model of quadruped robot movement and its movement algorithm.	26.05.2024 – 09.06.2024	Done
7.	Example of calculating the movement of a quadruped robot in a straight line. Analysis of the obtained results.	10.06.2024 – 12.06.2024	Done
8.	Preparation of an explanatory note, graphical materials, and presentation for the diploma project.	13.06.2024-19.06.2024	Done
9.	Submission of the qualification work for defense.	20.06.2024	Done

7. Task issue date: 01 “April” 2024.

Supervisor: _____ Tupitsin M.F.

(sign)

Task is taken for completion by: _____ Innokov O.O.

(sign)

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно-інтегрованих комплексів

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віктор СИНЄГЛАЗОВ

«_____» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Інноков Олександр Олегович

- 1. Тема кваліфікаційної роботи** «Алгоритм ходьби чотириноного робота».
- 2. Термін виконання роботи:** з 01 квітня 2024р. по 20 червня 2024 р.
- 3. Вихідні дані до роботи:** S. Talebi, M. Buehler and E. Papadopoulos. Towards Dynamic Step Climbing For A Quadruped Robot with Compliant Legs// Conference Paper in Proceedings - IEEE International Conference on Robotics and Automation · May 2013 DOI: 10.1109/ICRA.2013.6631035
- 4. Зміст пояснювальної записки:** 1) Огляд та аналіз алгоритмів руху рухомих роботів; 2) Опис особливостей руху крокуючих роботів; 3) Постановка завдання; 4) Кінематична модель руху чотириноного робота; 5) Розробка алгоритму руху робота; 6) Приклад розрахунку руху чотириноного робота по прямій лінії; 7) Аналіз отриманих результатів.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

Презентація в MicrosoftPowerPoint.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Ознайомлення з постановкою задачі кваліфікаційної роботи.	01.04.2024-04.04.2024	Виконано
2	Аналіз літературних джерел та інтернет ресурсів.	05.04.2024-24.04.2024	Виконано
3	Огляд та аналіз алгоритмів руху рухомих роботів.	25.04.2024-1.05.2024	Виконано
4	Опис особливостей руху крокуючих роботів.	2.05.2024-10.05.2024	Виконано
5	Постановка завдання;	11.05.2024-25.05.2024	Виконано
6	Кінематична модель руху чотириноного робота та алгоритм його руху.	26.05.2024-02.06.2024	Виконано
7	Приклад розрахунку руху чотириноного робота по прямій лінії. Аналіз отриманих результатів.	10.06.2024-12.06.2024	Виконано
8	Оформлення пояснювальної записки, графічних матеріалів та презентації до дипломного проекту.	10.06.2024-12.06.2024	Виконано
9	Подання кваліфікаційної роботи до захисту	13.06.2024	Виконано

7. Дата видачі завдання: “ 01 ” квітня 2024 р.

Керівник дипломної роботи _____

Тупіцин М.Ф.

(підпис керівника)

(П.І.Б.)

Завдання прийняв до виконання _____

Інноків О.О.

ABSTRACT

Explanatory note for the qualification work "Quadruped robot walking algorithm"

Research object – quadruped robot.

Research subject – walking algorithm of a quadruped robot.

The aim of the qualification work is to study and analyze the walking algorithm of a quadruped robot.

Research method – calculations, literature review, graphical modeling.

Theoretical research involved in-depth analysis of algorithms for the walking of a quadruped robot.

The research results showed which algorithms are used for the walking of a quadruped robot.

The results of the qualification work can be used to familiarize oneself with the walking algorithm of a quadruped robot and the data required to consider for this algorithm.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи "Алгоритм ходьби чотириноного робота "

Об'єкт дослідження – чотириногий робот.

Предмет дослідження – алгоритм ходьби чотириноного робота.

Мета кваліфікаційної роботи – Аналіз та розробка алгоритму ходьби чотириноного робота.

Метод дослідження - розрахунки, обробка літературних джерел, графічне моделювання.

Теоретичні дослідження полягали в глибокому аналізі алгоритмів ходьби чотириноного робота.

Результати досліджень показали, які алгоритми використовуються для ходьби чотириноного робота.

Результати кваліфікаційної роботи можуть бути використані для ознайомлення з алгоритмом ходьби чотириноного робота, які дані потрібні враховувати для цього алгоритму.

CONTENT

LIST OF ABRIVIATION

INTRODUCTION

CHAPTER I. WALKING PATTERN FOR QUADRUPED AS OBSERVER
ROBOT

1.1 Mechanical Design

1.2 Design of controller

1.3 Affordance Perception Model

1.4 Dynamic simulation

1.5 Walking Pattern

1.6 Experimental Result

CHAPTER II. WALKING ALGORITHM OF A FOUR-LEGGED ROBOT

2.1 A Four Legged Walking Robot with Obstacle Overcoming Capabilities

2.2 Leg Design and Reference System

2.3 Leg's Inverse Kinematics

2.4 Foot Stroke and Flight Trajectories

CHAPTER III. STATICALLY STABLE GAITS

3.1 Curved Paths

3.2 Obstacles Detection and Overcoming

3.3 Main Control Structure

3.4 Operational and joint coordinate in straight walking

3.5 Operational and joint coordinate in curved walking

3.6 Overcoming a step up obstacle

CONCLUSIONS

REFERENCES

APPLICATIONS

LIST OF ABRIVIATION

CNV-central nervous system

CPG-Central Pattern Generator

ODE- Open Dynamic Engine

AiDIN-Artificial Digigrade for Natural Environments

EoF-End of Effector

DH-Denavit-Hartenberg

CNS-central nervous system

PF-pattern forming

CoM point-center of mass

OOP-Object Oriented Programming

MVACS-Mars Volatiles and Climate Surveyor

INTRODUCTION

With the increase in climbing activities in the world, there has of course been an increase in the number of accidents. The most common are climbers getting lost, falling into toxic craters or entering caves with low oxygen levels. Large, inaccessible and dangerous places are the main obstacle for rescuers. The sooner the climber is found, the better the chances of establishing the likelihood of the climber's survival. In this graduate work, propose an observer robot that can help rescuers speed up their work when searching for victims in mountainous areas.

In recent years, there has been a great deal of research on robots, especially robots with legs, including bipedal, tripedal, quadrupedal and six-legged robots. The mechanical design of multi-legged robots certainly has advantages and disadvantages compared to wheeled robots. Multi-legged robots can successfully traverse uneven surfaces such as rocks, but they are not faster than wheeled robots. Of course, more leg configurations create their own problems, such as synchronizing the movement of each robot's legs to the desired position and coordinating their motion patterns.

Mobile robots with legs as a locomotion system have been the subject of much research. But at the same time, legs have a number of problems. In fact, research on legged robots focuses on everything related to the movement and coordination of the legs during the robot's navigation.

The main challenge in developing a legged robot is the coordination of the legs' degrees of freedom. It is the coordination of a large number of degrees of freedom that must be triggered in a simple movement of the robot from point A to point B. Precise coordination of walking movements, the precise coordination of the walking motion is to ensure that the dynamic forces generated during navigation keep the robot in balance. This is a major challenge. These limitations can be overcome by modeling statically. The dynamic balance of forces due to the robot's movements can be overcome by statically modeling it.

Collisions with obstacles or ground irregularities can cause the robot to lose its balance and fall due to lack of additional support points or collision of a leg with some height. Robot navigation requires obstacles to be detected and avoided while guiding the robot to the goal. Known and unknown, moving and stationary obstacles in the robot's path must be considered, as well as the need for detection and avoidance strategies.

To operate safely in public environments, many successful systems use sensor-based modules for collision avoidance to control mobile robots. The main control structure is sensory response: sensor readings are continuously analyzed to determine collision-free motion.

Path planning for wheeled and legged mobile robots has attracted a great deal of interest, and several studies have been conducted on developing skills to detect and overcome obstacles and to enable the robot to walk on irregular surfaces with obstacles. In addition to navigation skills, the robot's balancing strategy is also required to fulfill the task of overcoming or avoiding obstacles in its path.

Relevance of work: The use of mobile robots is justified in many areas of application in unstructured environments where a high degree of autonomy is required. This necessary autonomous or intelligent behavior is the result of intensive research today. Motion control of mobile robots without fixed wheels has been an important research topic for the past few years. However, in the absence of the necessary degrees of freedom of a mobile robot, ensuring its controllability is a very important and difficult task. Due to its non-holonomic status, the CVM cannot be stabilized at a point using the usual patterns of smooth maneuvering, as in holonomic robots. Therefore, the problem of robot stabilization is still an understudied and relevant topic. The relevance of the research topic is determined by the need to create more advanced control systems for wheeled robots. Research goals and objectives

The purpose of this work is to develop an algorithm and design a software module for the stabilization of a wheeled robot based on a control system using a universal kinematic model and a contour motion control algorithm.

- In order to achieve the set goal, the following main issues must be resolved.
- Classification of mobile robots;
- Analyze and classify methods of managing mobile robots; and
- Classify the subtasks of stabilization of mobile robots;
- Analyze wheeled robot control models;
- Development of trajectory generation models for mobile robots;
- Develop stabilization models for mobile robots;
- Development of stabilization algorithms for mobile robots;
- Development of mobile robot control schemes; Development of mobile robot control schemes;
- Develop schemes for connecting software modules;
- Carry out simulation modeling of the developed models and algorithms of mobile robot stabilization;
- Develop recommendations for the protection of software and hardware complexes.

Research object: robot management system.

Research subject: algorithms and models of robot stabilization.

CHAPTER I

WALKING PATTERN FOR QUADRUPED AS OBSERVER ROBOT

1.1. Mechanical Design

Despite the rapid development of robotics, living organisms are in many ways superior to existing robots. Therefore, in order to control robots, it is necessary to understand the principles underlying the movement and behavior of living organisms. Mimicking living organisms is currently one of the global trends in robotics innovation. Since biological systems can provide many useful ideas for robot control, this is recognized as one of the most feasible ways to develop robots. In recent years, biologists as well as roboticists have proposed innovative ideas for the control of walking robotic systems. Among the various ideas, mimicking the rhythmic movements of animals is one of the most promising ways to control walking robot systems. By working on this, the motion of a walking robot can be approximated to that of a real animal [1].

Recent studies by neurobiologists have shown that the Central Pattern Generator (CPG), part of the nervous system of the animal brain, produces rhythmic movements when in motion. The CPG can produce rhythmic movements even without sensory feedback. During the investigation of CPGs, the study of real animal brains and neurons in the body has led to the development of a number of new approaches, including the McCulloch and Pitts neuron, the Leaky integrating neuron, the Matsuoka neuron, the Leaky integrating neuron, the Matsuoka neuron and the Leaky integrating neuron, the Matsuoka neuron, and many other neural models have been proposed. Algorithms inspired by rhythmic pattern generators have also been studied to develop methods for robot control. Although these neural models show how animal movements can be controlled, the results of their application to robotic systems are very different from those of real biological systems, because the multiple neurons introduced so far by robotic systems cannot simply mimic animal movements.

Oncontrolling a quadruped walking robot, various aspects should be taken into account. In this research, several critical considerations are given for the design of a quadruped walking robot controller. Mainly, the basic principles of the quadrupedal locomotion controller are analyzed in terms of a biomimetic point of view. By the observation of the gravity load receptor, stimulus-reaction mechanism of animals are investigated using the stances on walking and posture control. The controller design factors are feedback information from several receptors and stimulus-reaction based on biological sensory system analysis. The proposed method is validated via dynamic simulation using Open Dynamic Engine(abbreviated as ODE) and then, it is experimentally proved by applying to aqua druped walking robot, called AiDIN [2].



Fig. 1.1 AiDIN (Artificial Digigrade for Natural Environments)

Overview of The Platform : UdieBot

UdieBot is a quadruped robot with three degrees of freedom in each leg, as shown in Fig. 1.2 Each degree of freedom represents a joint that the robot has. An actuator is needed to move the joints. In this robot, servo motors are used as actuators. Therefore, 12 servo motors are used in the legs of the whole robot, which is the same as the number of joints. The same type of servo motor (Hitec HS-7954SH) is also used to synchronize the joint movements of the whole leg.

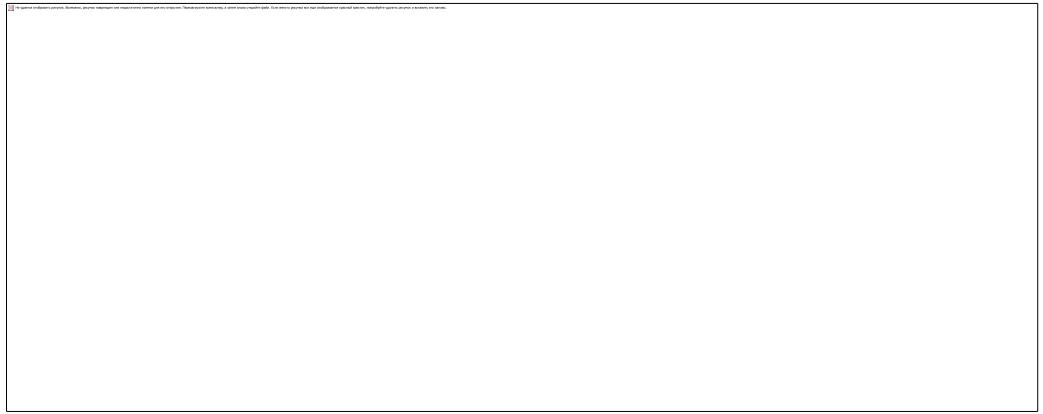


Fig. 1.2 Quadruped as observer robot

The robot weighs 1.56 Kg, is 200 mm high, 350 mm long and 350 mm wide. The robot has 12 DOFs and each leg has 3 DOFs. Other specifications such as weight, body dimensions, power supply and type of controller used can be found in Table 1.1. There are two controllers, both using STM32F4. The first controller is used to process the whole system, including data communication, receiving and processing sensor data. The second controller is used to control the movement and navigation of the robot [3].

Table 1.1 Robot specifications.

Mechanical Design

The first step in designing a robot is to solve the mechanical aspects. The mechanical design of Udibot is a quadruped robot with three degrees of freedom in each leg, as shown in Fig. 1.2 in order to control Udibot's movements such as

forward motion, backward motion, right shift, left shift and rotation, kinematic modeling must first be completed. For this reason, forward kinematics must first be completed and inverse kinematics equations must be created. The details of the mechanical design, forward and inverse kinematics of the whole robot consisting of legs have been done before [4].

Legs

The leg mechanism of Udiebot consists of three servomotors: servo hip (1), servo femur (2) and servo tibia (3). Details are shown in Table 1.2. The hip servo coxa and servo femur are combined in a single frame. The tibial servo is connected to the knee (see Fig. 1.3).

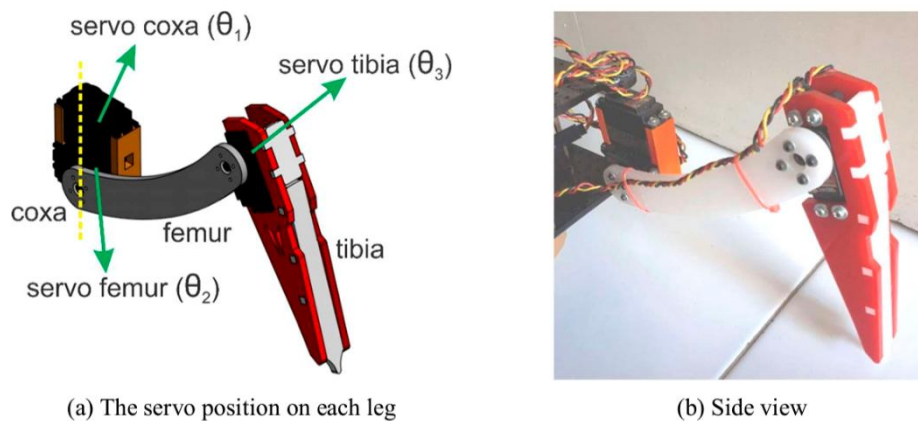


Fig. 1.3 Robot's leg.

Most of the mechanisms were made using a 3D printer, while others were made from acrylic sheet; the 3D printed frame is the orange part, while the other frames were cut from acrylic sheet. Tables 1.2 and 1.3 show the connection lengths of Udiebot's legs and the position of each actuator.

Table 1.2. Leg parameters

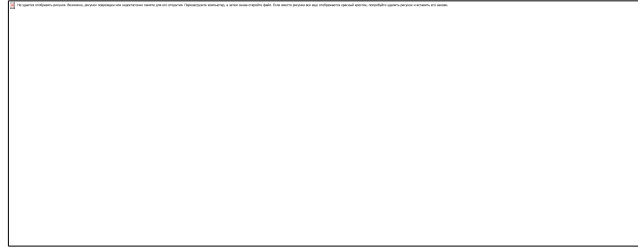
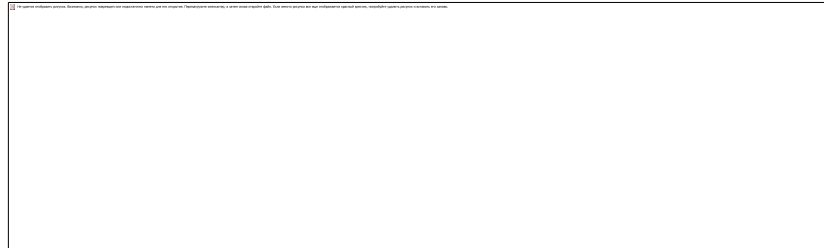


Table 1.3. Actuators and DOF.



Udiebot's legs have two actuators called servomotors in the femur and hip joint. On the other hand, the knee (step) has a single actuator called the tibia servomotor. Since the robot has four legs, the robot has a total of 12 degrees of freedom.

Forward Kinematics

Forward kinematics is used to obtain the position coordinates of each joint/joint and the equations of motion that generate the EoF (End of Effector) attitude of the robot. Denavit-Hartenberg (DH) parameters are used for each homogeneous matrix state.



(1.1)



(1.2)

Inverse Kinematics

Inverse kinematics is often used to move manipulator robots relative to the position of desired coordinate points. To move a robot leg, the position of the coordinates in Cartesian space must first be defined. This process requires input data

in the form of coordinate points (x, y, z) and produces output data in the form of angle values for each robot leg joint [5].

$$legLength = \sqrt{posX^2 + posY^2} \quad (1.3)$$

$$Zlength = \sqrt{posZ^2 + legLength^2} \quad (1.4)$$

$$\theta_A = \tan^{-1}((legLength - coxaLength)/posY) \quad (1.5)$$

$$J = \cos^{-1}\left(\frac{Zlength^2 - femurLength^2 - tibiaLength^2}{-2.FemurLength.Zlength}\right) \quad (1.6)$$

$$K = \cos^{-1}\left(\frac{femurLength^2 + tibiaLength^2 - Zlength^2}{-2.FemurLength.tibiaLength}\right) \quad (1.7)$$

$$\theta_1 = \tan^{-1}(posX, posY) \quad (1.8)$$

$$\theta_2 = J + \theta_A \quad (1.9)$$

$$\theta_3 = 90^\circ - (K) \quad (1.10)$$

Equation 3-7 is used to find the value of supporting parameters. While Equation 8-10 is the main formula that calculate final value for θ_1 , θ_2 and θ_3 .

1.2. Desig of controller

Animals normally move by the action of muscles. When muscles produce a specific movement, the act of control is carried out through two different loops. In the first cycle, commands from the brain are transmitted to the muscles through the central nervous system (CNS). In the second cycle, the muscle responds to external stimuli that it cannot predict or recognize. The stimuli are detected by receptors such as proprioceptors and exoreceptors. Its structure is highly complex, but movement can be simplified by competing actions such as flexion and extension; the composition of the GLC varies from system to system, but the basic structure can be shared [6].



Fig. 1.4 Junction control section consisting of four parts: oscillation section, stimulation section, threshold detection section, and PD control section.

As shown in Fig. 1.4, the GLC of the quadruped robot consists of four parts: an oscillator, a stimulator, a threshold detector and a PD controller. Moreover, the proposed controller has two loops (overshoot and undershoot) that are excited separately according to the threshold detector. If the information from sensory feedback (called charge receptors in biology) such as force sensors, touch sensors, gyro sensors, etc. exceeds a fixed threshold (safe zone), the stimulator is activated according to the overshoot value: continuous stimulation is provided when the feedback is outside the safe zone, guiding the robot to a safe state. This stimulus can

be expressed as the number of pulses continuously added to the leg until the safety requirements are met. If the feedback is a large perturbation, an intermittent stimulus, usually a large pulse, is generated to prompt an immediate response. In the case of failure to exceed, the stimulus is disabled and the robot operates only under the desired trajectory input [7].

This paper proposes a new rhythmic pattern generator that is simple and easy to implement. The study starts by observing the movement of a one-step class.

As shown in Fig. 1.5, when a one-step class walks in a flat environment, the one-step class repeats the play and support phases at approximately equal intervals.



Fig. 1.5 Relationship between joint angles of each leg

In addition, the joint angles of each leg have a competitive motion that keeps the phase between each joint angle constant. In particular, if the relationship between the joint angles of each leg can be understood, this property can be applied to the motion control of a quadrupedal walking robot. Although many researchers have introduced the SWvs.ST ratio as a function of walking speed, it is not easy to apply this idea to quadruped robots. Previous studies cannot measure and normalize the walking pattern of a digital walking robot, as shown in Table 1.4 [8].

Table 1.4 swing mode (SW) vs. support mode (ST) (1 cycle)



A Bezier curve is used as a typical average. Here κ is expressed as a function of time $\in[0,1]$.

$$\kappa_i(t) = \sum_{k=0}^n \frac{n!}{(n-k)!k!} P_k (1-t)^{n-k} t^k \quad (1.11)$$

here P_k represents the k th control point. The trajectories of the joints in Fig. 3 can be represented by three Bezier curves with four and six control points respectively. From Equation (1), the rhythm model of the quadruped robot can be designed.

The rhythmic pattern generator κ generates a motion pattern. In other words, as shown in Fig. 1.3, κ always produces a motion pattern for any input value.

This is different from the pattern generators introduced so far. As shown in Fig. 1.6, the overall controller of the quadruped robot consists of a four-joint controller working synchronously with the gait pattern determination switch.

The desired final position x_i of each leg is given by the exciter and oscillator as follows.

$$x_i = \kappa_i + \delta x_C + \delta x_I \quad (1.12)$$

and transferred to the joint space through the inverse kinematics calculation process. The desired joint angles θ_{ij} of leg i and joint j are then controlled by the PD controller at the joint level.

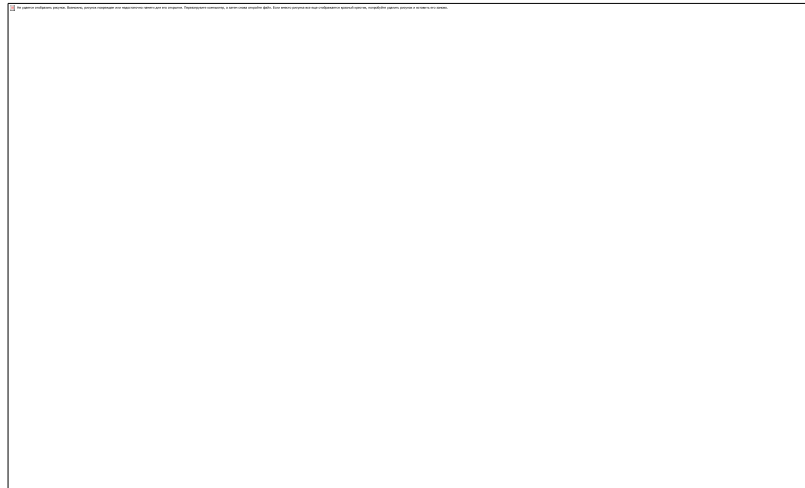


Fig. 1.6 Overview of motion controllers

Robot equipment design

Robots suitable for use in disaster situations must be able to navigate all types of rough terrain, slopes and natural terrain (grass, rough terrain), pass through narrow spaces and climb stairs and vertical ladders . If we turn to the animal world for inspiration, the feline family (Felidae) stands out for its omnipotence. Cats can withstand many harsh environmental conditions. They can swim, are agile and climb trees. Cats are the optimal prototype for a flexible four-legged robot. Our robot cat is shown in Fig. 1.7 [9].

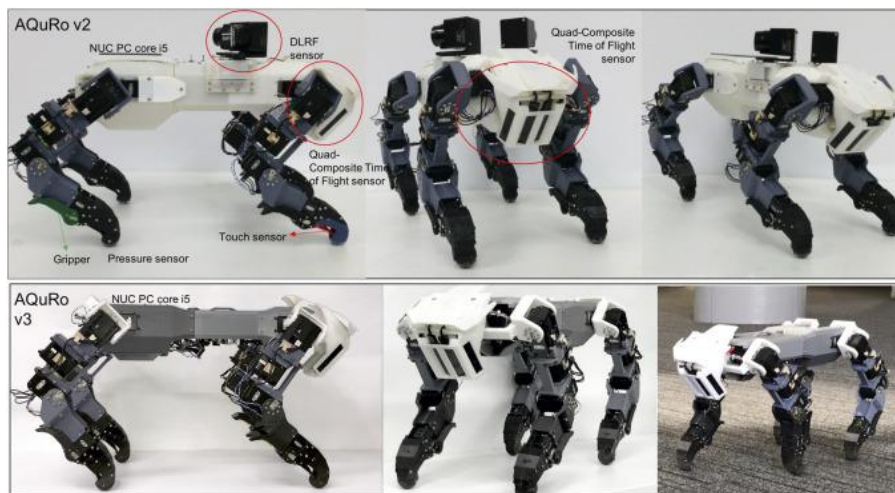
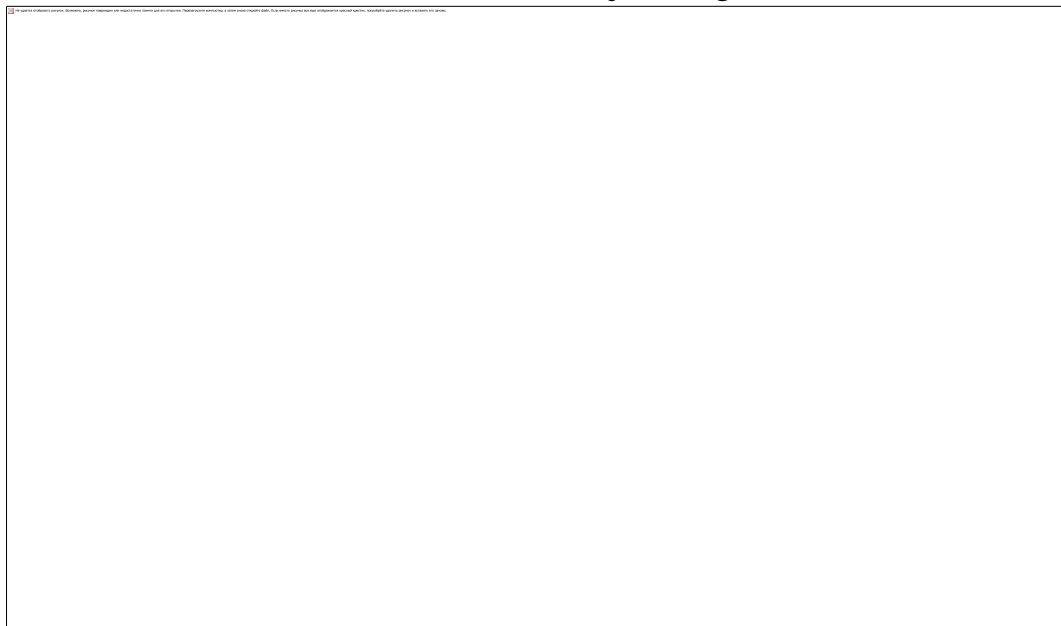


Fig. 1.7 A four-legged robot. The AQuRo v2 is equipped with a DLRF sensor, while the AQuRo v3 has a thinner body without a DLRF sensor.

Mechanical design

The size of the proposed four-legged robot is close to the size of an adult cat: 25 cm wide x 60 cm long x 30 cm high; the robot weighs about 7 kg. Fig. 1.8 The front limbs of the robot imitate the structure of the front limbs of cats without ankle joints. There are only two joints: shoulder and elbow. Three actuators are connected to the design of the ball joint of the shoulder and one actuator is connected to the design of the hinged joint of the elbow. To design the rear legs of the robot, we took into account the rhythmic movements of the cat. In such movements, the proximal and distal segments of the legs maintain a relative angular posture during most of the cycle, and the angular joints deviate only at the beginning of the takeoff (Witte et al.). So we simplified the rear leg by eliminating the knee joint and allowing direct integration of the ankle and hip joints. The legs are shown in Fig. 3. Each leg has five degrees of freedom, one of which is used as a gripper joint. The length of the tibia is 175 mm, and the femur is 145 mm. The parameters of the Denavit-Hartenberg robot are listed in Table 1.5 [10].

Table 1.5 DH Table of the joint leg structure.



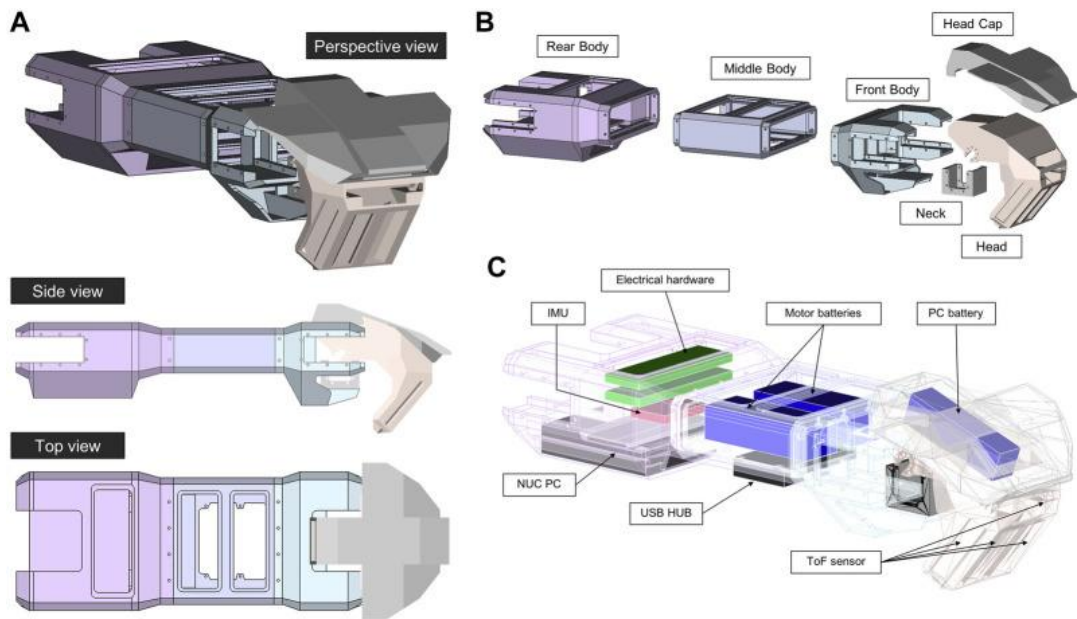


Fig. 1.8 The robot body (torso and head). (A) Orthographic projections. (B) Interior hardware placements. (C) Exploded parts.

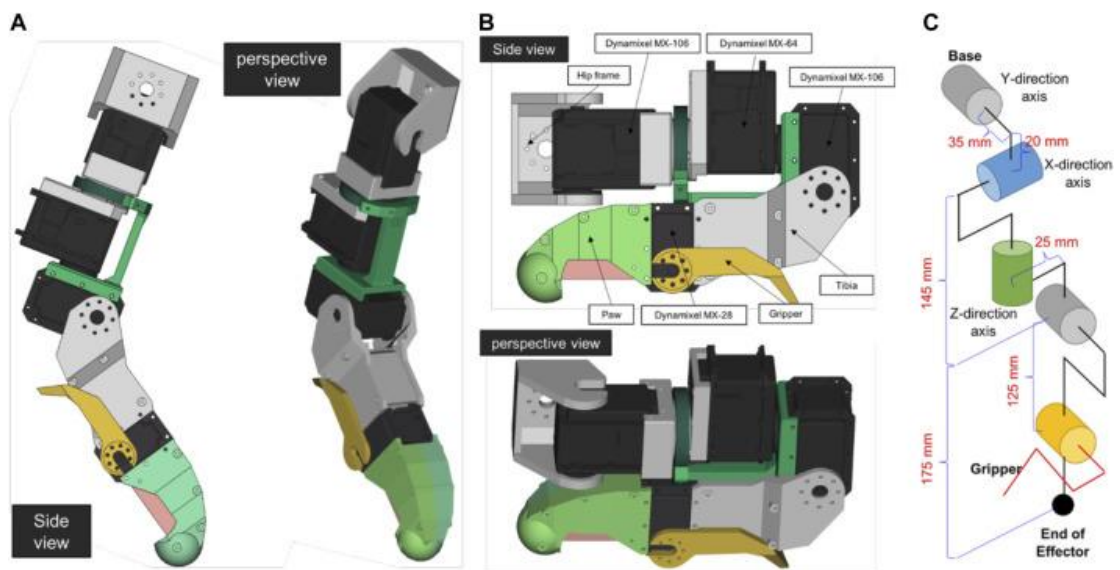


Fig. 1.9 The design of Leg (A) flexed, (B) extended. (C) Leg actualization.

Robot Body The general shape of the robot body is shown in Fig. 1.7.

The body is 3D printed using polylactic acid (PLA). The body of the robot consists of three parts: back, middle and front. The rear legs are mounted at the back and contain the NUC computer, IMU sensors and electrical equipment. Two motor batteries and a USB hub are placed in the middle part. The front part serves for fastening on the neck and head. The main section houses the PC battery [11].

End sensor The end sensor is designed to support agile movements, such as walking over rough terrain or climbing vertical stairs. The end effectors must also measure the ground reaction force and meet size restrictions. Cats use their claws to grasp rock walls, trees, and poles while climbing, and Shiquan et al. developed an end effect with a dense array of microspines for rock climbing (Wang et al., 2019). However, this is a larger area than is suitable and necessary for our proposed work. In addition, cats grasp with two coordinated limbs. Cats also have difficulty climbing vertical stairs [12].

Humans and apes, on the other hand, have hands to grasp and hang on stair rails. However, manual mechanisms for such movements require high torques and, accordingly, large servo motors. We have simplified this task by using a hook-shaped end effector that does not require an actuator. The design is shown in Fig. 1.10. The rounded end is designed to facilitate attachment of the foot and no drive is required. In addition, the sensor design placed a force-sensing resistor (FSR) between the top and bottom of the forefoot. The switch in the hook cavity acts as a sensor that determines whether the front part of the foot is engaged in the step. Behind the front foot are movable claws for grasping and supporting the hind foot. Driven by low-torque servo motors, these claws help prevent slipping [13].

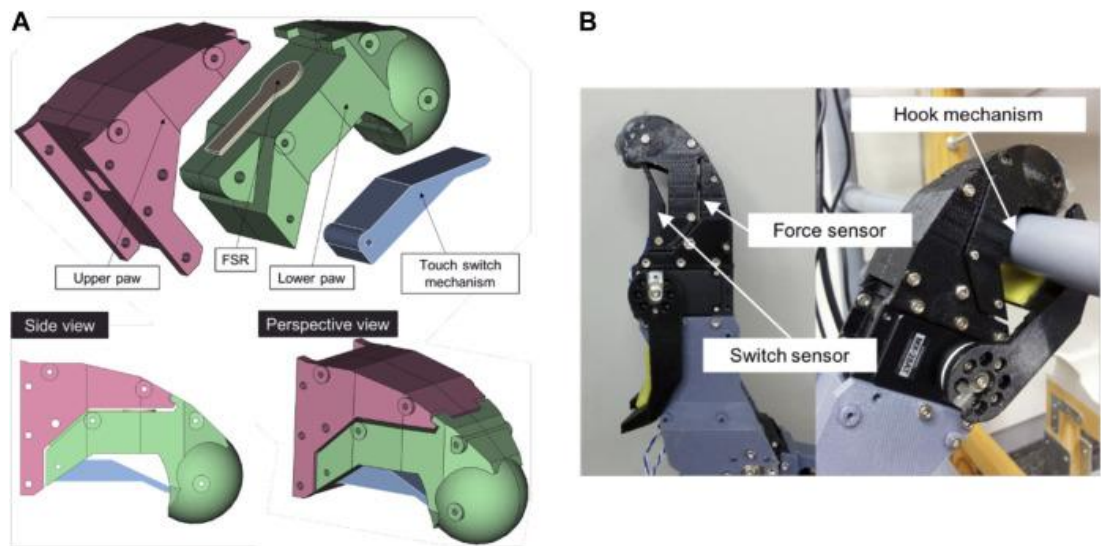


Fig. 1.10 The end effector. (A) CAD drawings (B) The 3D-printed effector.

Sensory System

The robot cat is equipped with a variety of sensors representing external and internal sensory receptors. As external receptors, we created a four-component time-of-flight sensor to sense the environment in front of the robot, and a dual laser rangefinder to observe a larger area. As interceptors, we installed a force sensor (resistance sensor) and a touch sensor (microswitch) on each leg, as well as an inertial measurement unit (IMU) inside the body. The IMU module uses the NG-IMU shown in Table 1.6.

Table 1.6. NG-IMU sensor specifications.

Model	NG-IMU
Sensors	Gyroscope, Accelerometer, Magnetometer, Pressure, Humidity
Update Rate	400 Hz
Static Accuracy (pitch/Roll)	<1 [deg] RMS
Static Accuracy (Heading)	<2 [deg] RMS
Communication	USB, Serial, WiFi
Size and weight	50 × 33 × 8 [mm], 10 [gram]

Four-component time-of-flight sensor This sensor is mounted on the robot's head. It consists of four CamBoard pico flexx ToF composite sensors with pmd, as shown in Table 1.7. The structure of the composite sensor is shown in Fig. 1.11. This design, combined with the geometry of the robot head, provides a large field of view that minimizes the number of actuators needed on the robot neck . Therefore, the neck is equipped with one actuator that rotates in the sagittal plane; The CAD design and photo can be seen in Fig. 1.12 [14].

Table 1.7. Time-of-flight sensor specifications.



Table 1.8 LRF sensor specifications.

--

Electrical equipment The robot has a hardware configuration for processing internal and external sensory information. The hardware configuration is shown in Fig. 1.13. An ATmega 8 microcontroller is used as a sub-controller to pre-process the internal sensory inputs from the force sensors, touch sensors and IMU; NUC PC Core i3 is the main system for processing various advanced systems such as sensing, motor control, communication and interface. It functions as a controller [15].

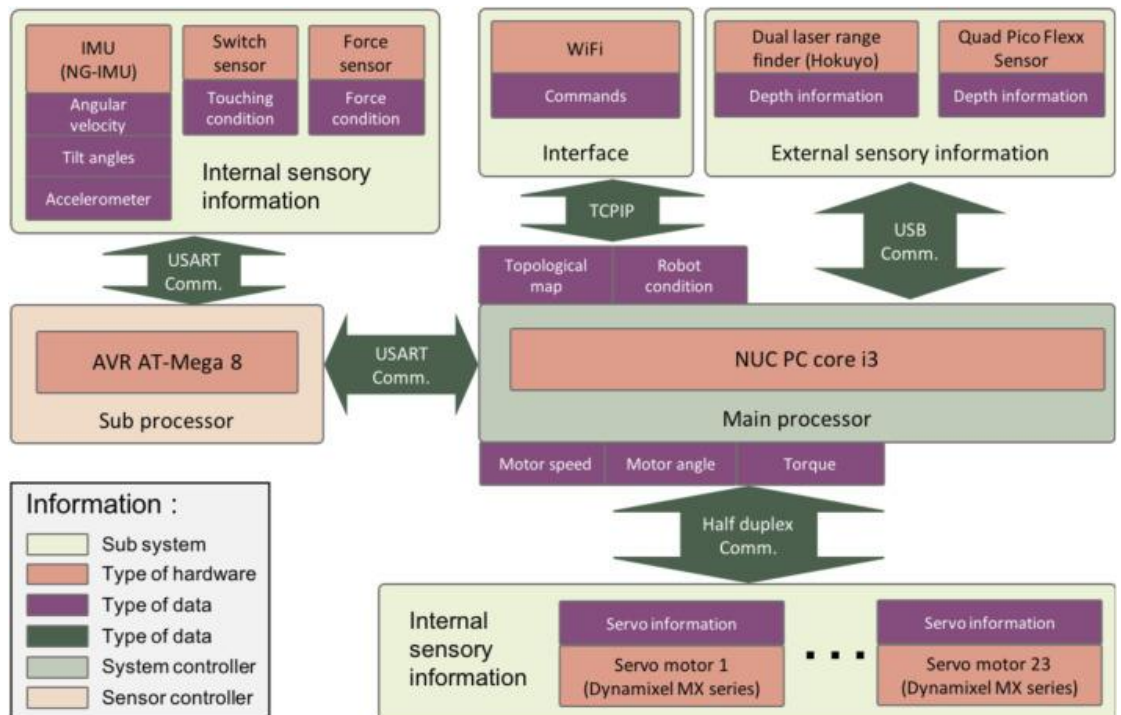


Fig. 1.13 Structure of the electrical system.

The sub-controller processes the analog signals from the force, touch, and IMU sensors through the analog-to-digital converter inputs. The data stream is then transferred to the main controller via a USB connection. All external sensor information is also transmitted via the USB connection. The master controller generates digital control signals for all servo motors: 12 Dynamixel MX-106 servo motors, 4 Dynamixel MX-64 servo motors, and 7 Dynamixel MX-28 servo motors.

The electrical system is powered by two 2700 mAh (14.8 V) 4-cell Li-Polymer batteries and one 2200 mAh (14.8 V) 4-cell Li-Polymer battery. These batteries can power the robot for about 15 minutes.

Movement-Related Capabilities

We implement robot capabilities related to movement by integrating external and internal sensory information. This integration allows external sensory information to be reflected in motor behavior with a short adaptation time, as in animals. For example, if an obstacle suddenly appears during the walk-swing phase, the swing changes accordingly and the leg enters the safe zone. This mechanism demonstrates the importance of external sensory information (in this case, visual) in motor behavior [16].

To use depth information and 3D point cloud data as external sensory information, the processing system of our robot includes 1) a dynamic density topological map generator using an attention model, 2) a physical shape detector using topological maps, and 3) a motion model based on neural network.

1.3. Affordance Perception Model

The concept of appropriateness comes from Gibson in ecopsychology, and Turvey defines appropriateness as a dispositional characteristic of the environment. The actor's activity or disposition complements what the environment provides. Dispositions contain important details that define the potential behavior and capabilities of an actor. Therefore, differences in the arrangement of robots can lead to different perceptions of accessibility. The goal of our proposed adaptation perception model is to find the appropriate integration between the environmental conditions and the possible behavior of the robot. We created a fitness sensing system for robot movement, stair detection, and capture [17].

State Sensing for Locomotion In locomotion systems, active behavior is controlled by opportunity sensing. Thus, positive behavior is generated according to the received information about the state of health. In our model, capabilities are revealed by exploring the plane of the topological map generated by a dynamically growing density neural gas (DD-GNG). The possibilities we are interested in are the horizontal (or near-horizontal) planes that the robot can step on. They are found by calculating the slope of the planes. The normal vectors of the triangular surfaces of the topological structure are calculated using equation (1), as shown in Fig. 1.14.

$$\mathbf{N} = \frac{(\mathbf{n}_0 - \mathbf{n}_1) \times (\mathbf{n}_0 - \mathbf{n}_2)}{\|(\mathbf{n}_0 - \mathbf{n}_1) \times (\mathbf{n}_0 - \mathbf{n}_2)\|} \quad (1.3.1)$$

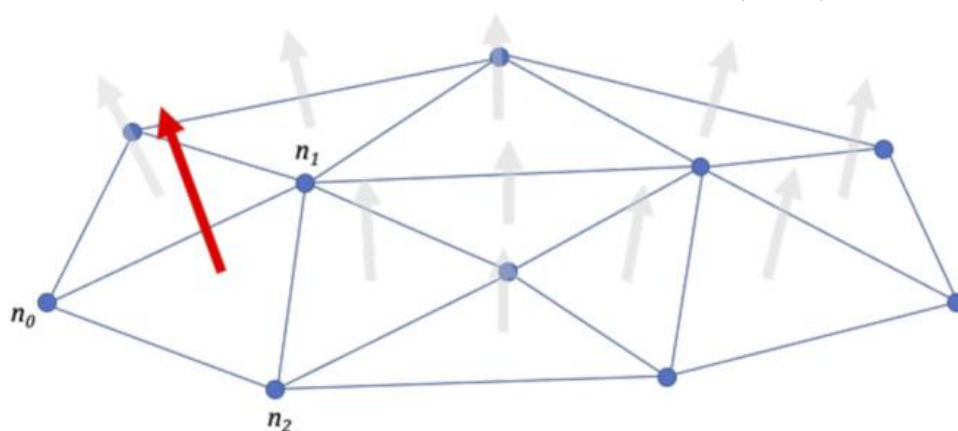
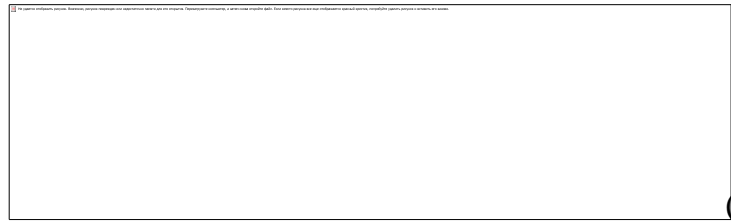


Fig. 1.14 Calculating the normal vector of a plane triangle.

Equation (2) should be used to calculate the slope of the plane (γ_i) on all surrounding surfaces. The steepness of the planes can be taken into account to calculate the safety factor of the step.



(1.13)

The model is designed for low-cost detection of vertical stairs in real time based on 3D point cloud data; the DD-GNG output is used as input to the fitness model. Feature extraction is necessary to identify suspicious artifacts for further processing. The detection of the steps of the vertical stairs is then processed using a layer-in layer-out system. Thus, a staircase detection system represents a staircase as a series of nodes and edges. Then the positions that can be grasped are determined, taking into account the shape of the robot's body [18].

Capture Match Detection This match detection process aims to identify positions on an object that can be captured. The process creates a seven-dimensional representation of the position that can be captured: (3D position, 3D rotation, object diameter). To detect the target object, an RGB camera is placed on top of the robot's four-component rotation sensor. The target object is detected using a computer vision algorithm. Topological structures are then generated using the proposed DD-GNG. The density of the topological structure is concentrated on the target object. On the basis of the multiplication-subtraction operation, the capture probabilities are determined based on the information of the topological map and the robot capture program. The capture probabilities are then ranked from "best" to "worst" in the defined capture decisions [19].

Locomotion Model

Our locomotion model is a response to the challenges of modern CPG development in quadruped locomotion research. We propose an efficient and reliable CPG model that dynamically integrates sensory feedback to generate

different types of gait and can compensate for leg dysfunction without significantly increasing the number of parameters. The model has two feedback mechanisms based on sensory-motor coordination and sensory-motor coordination. In the first feedback mechanism, sensory feedback is used to modulate CPG modulation. This occurs by providing feedback to the rhythm generating (RG) neurons with an intrinsic receptive signal representing force application and the oscillatory phase of the leg. This feedback is reduced by a second feedback mechanism when the leg is injured. Nociceptor neurons in the injured leg send signals to the RG that modify the effects of other sensory signals in the same leg. In addition, they integrate functions at the supramedullary level generated by motor functions and cognitive information. Our general model simulates a bottom-up influence from attentional mechanisms to muscle activation under the influence of visual information. Our model considers the problem of short-term adaptation to the sudden perception of an obstacle Table 1.9 .

Table 1.9. Table of parameters.



We developed a unified CPG model in which each RG neuron represents a leg movement pattern and each pattern forming (PF) neuron represents muscle

activation. Since we use four muscles in the leg (hip flexors and knee extensors), each limb structure in the CPG network consists of one RG neuron and four PF neurons. In general, our model uses two CPGs, one for the forelimb and one for the hindlimb. The general design of the CPG is shown in Fig. 1.15, where we have extended the CPG model from the previous model presented in (Saputra et al., 2020b). The Matsuoka neural oscillator model is used to generate a dynamic signal; the internal state of RG neurons can be seen in the following equation:

$$\tau \frac{d}{dt} x_i = \left(v_i - x_i - \sum_{j=1}^n \mathbf{w}_{(RG,ij)} y_j + \alpha_i - b v_i \right) (\tau_f S_{STIM}) \quad (1.14)$$

$$T \frac{d}{dt} v_i = (y_i - v_i) (\tau_f S_{STIM}) \quad (1.15)$$

$$\alpha_i = \alpha_{i,0} + \sum_{j=1}^n \mathbf{w}_{(FR,ij)} F_i N_j - \sum_{j=1}^n (G_{STIM,A} (S_i) (\mathbf{w}_{(SR,ij)} N_j) - \sum_{j=1}^n (G_{STIM,B} (S_i) \mathbf{w}_{(NS,ij)} N_j)) \quad (1.16)$$

$$G_{STIM,A} = 3 / (1 + \exp(-8 S_{STIM} + 15)) \quad (1.17)$$

$$G_{STIM,B} = 2 \exp(\log(0.5) (2 S_{STIM} - 4)^2) \quad (1.18)$$

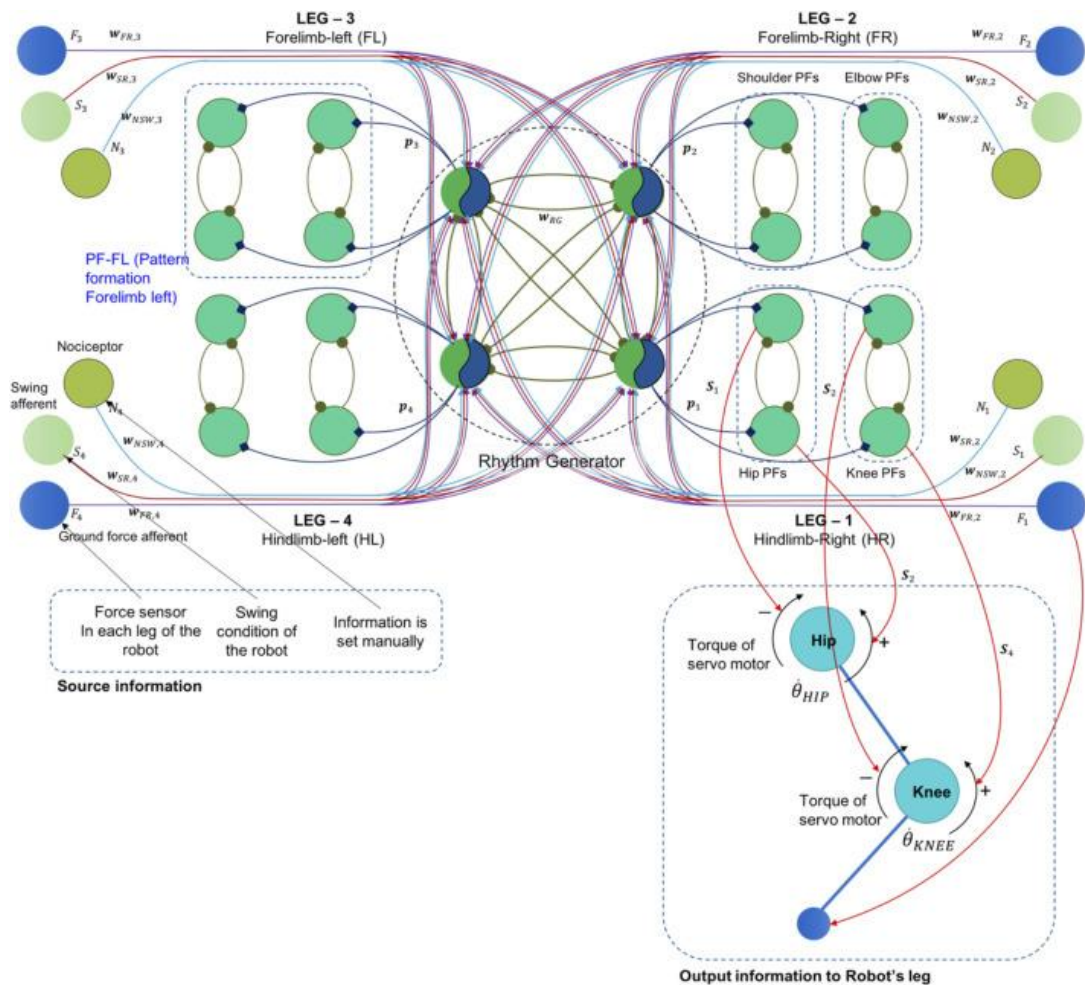


Fig. 1.15 The single-rhythm CPG model with a two-layered CPG. The rhythm generator neurons received feedback signals from a force sensor, a pain receptor, and a swing sensor in each leg.

Signals from RG neurons are transmitted to PF neurons, which generate spikes for certain kick movements; the spike signal (P_i) of the PF neuron is calculated by Equation 6, and the reference signal (h_i^{ref}) by Equation 7, where R is the subtraction constant parameter, $\gamma \text{ ref } h_i^{ref}$. The source code of the CPG model is also included in Appendix S3.

$$p_i(t) = \begin{cases} 1 & \text{if } (y_i + h_i^{ref}(t)) > q \\ 0 & \text{Otherwise} \end{cases} \quad (1.20)$$

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \Delta h_i^{ref}(t-1) - R & \text{if } (P_i(t-1) = 1) \\ \gamma^{ref} \Delta h_i^{ref}(t-1) & \text{Otherwise} \end{cases} \quad (1.21)$$

$$PF_{i,k}(t) = e^{\left(c_1 \left(\frac{|P_{i,k}(t)-\mu|}{\mu \Delta w}\right)\right)}, P_{i,k}(t) = P_{i,k}(t) + p_i(t) \quad (1.22)$$

In this process, RG neurons have a rhythmic signal and generate a burst signal to PF neurons. The PF parameter $i,k(t)$ in Equation 8 is the signal generated by the k -th PF neuron in the i -th leg. This activates muscle stimulation (Saputra et al., 2020a), which has been described in previous studies (Saputra et al., 2020b). The output of the muscle excitation (S_i) is transformed into the direction of the torque of the robot's leg servos. Fig. 10 shows the connection information: the torque of one servomotor is controlled in different directions by two muscle stimuli; the flexor stimulus is driven in the CW direction and the extensor stimulus is driven in the CCW direction. From Fig. 10, you can roughly calculate the total torque and angular velocity of the servo according to equations (9) and (10) [20].

$$\tau_{HIP}(t) = rS_1(t) + rS_2(t) \quad (1.23)$$

$$\dot{\theta}_{HIP}(t) = \dot{\theta}_{HIP}(t-1) + (S_1(t) - rS_2(t)) / r \quad (1.24)$$

1.4. Dynamic simulation

To test the proposed idea, a dynamic simulation with a four-legged walking robot model was performed. In this study, a walking quadruped robot simulator was developed using the Open Dynamic Engine (ODE, <http://www.ode.org>) in a Linux environment.

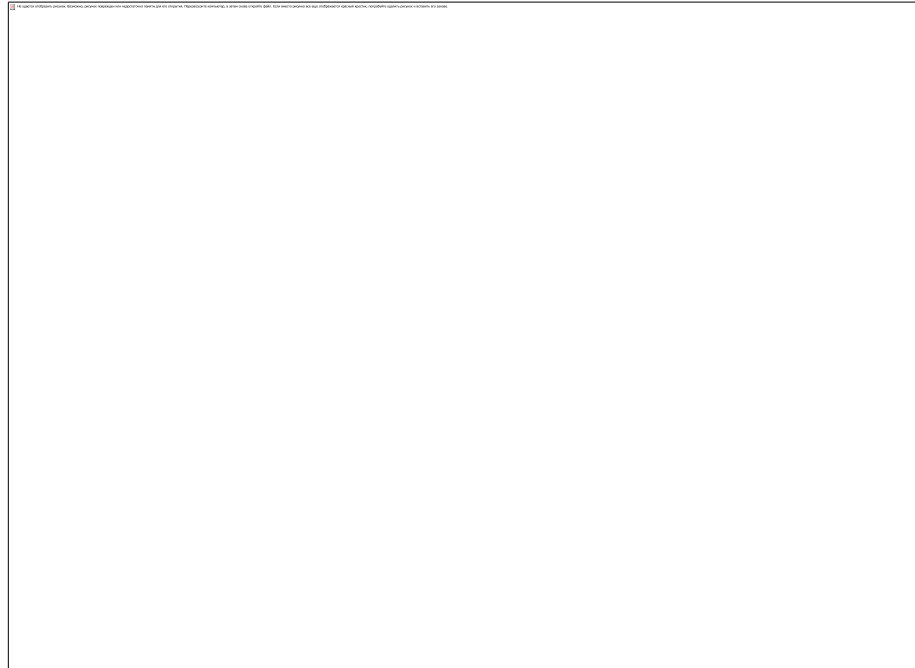


Fig. 1.16 kinematic structure for simulation

The model of the quadruped robot is shown in Fig. 1.16 and its kinematic parameters are given in Table 1.10. The model consists of 29 rigid parts connected by 18 joints (rotational and fixed joints). The skeleton of the model contains all physical information in parameterized form. The limbs of the model resemble the leg structure of the digigrade and the lengths of the limbs were determined accordingly. In particular, the forelimbs have three degrees of freedom: the scapular joint (A), the knee joint (B) and the clavicle joint (G). The collarbone joint is necessary to adjust the width between the supporting legs, whose vertebrae carry a weight of 1 kg, as shown in Fig. 1.16. Geometric similarity with digigrade was used to determine the dimensions of the robot. In the present simulation the length of the robot, from the scapula joint to the hip joint, was taken as the model height

(approximately 30 mm). As in many quadrupedal mammals, the length between the scapulocoracoid joints is longer than the pelvis [22].

Table 1.10. Physical properties of the simulation model

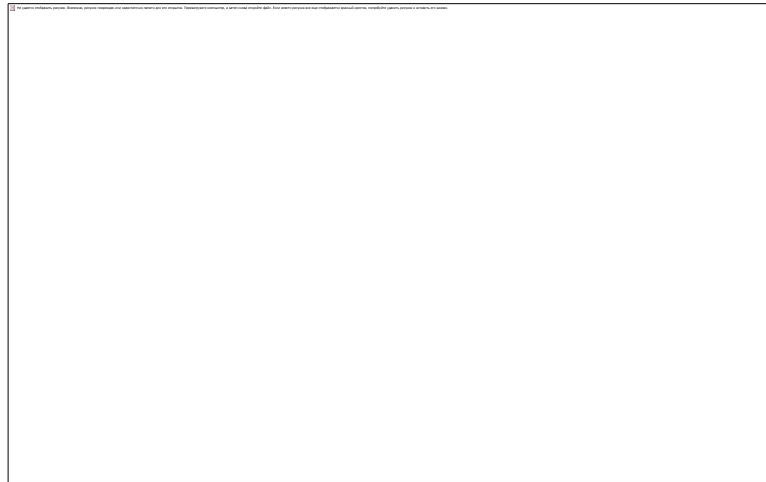


Fig. 1.17 (a) Plane, (b) sloping surface with 10° slope and 10° slope angle, (c) plane with 50 cm x 3 cm x 500 cm steps at 50 cm intervals, (d) rough terrain

As shown in Fig. 1.17, the simulation environment can be selected from flat, sloping, stepped flat and hilly terrain, and the simulation results are collected respectively. To objectively compare the energy consumption of the quadruped robot with the proposed controller, a measure of motion energy consumption called specific resistance is used.

$$\boxed{} \quad (1.25)$$

Where E is the specific resistance, E is the total mechanical energy consumption for the linear displacement of L , m is the total mass of the robot and g is the gravitational constant. In equation (3), a conservative measure of energy consumption can be expressed as the integral of the absolute value of the instantaneous shaft power. In other words, positive and negative torque are considered as the total mechanical energy consumption .

$$\boxed{} \quad (1.26)$$

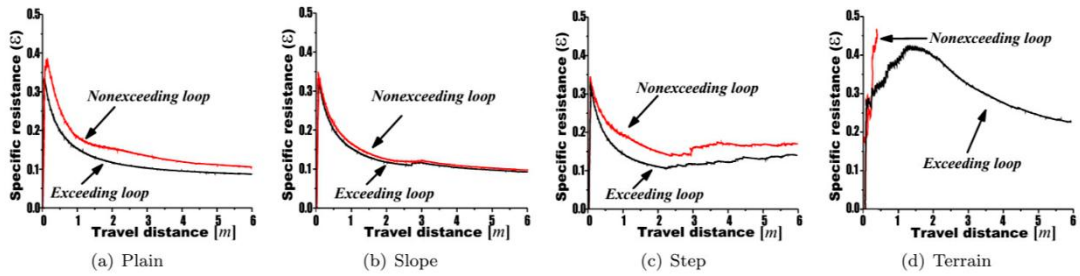


Fig. 1.18 Specific resistance versus distance

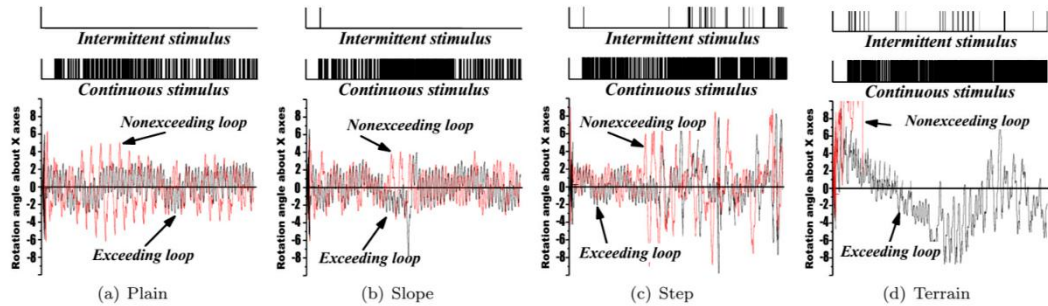


Fig. 1.19 where τ_{ij} and ω_{ij} are the torque and angular velocity of the leg joints.

Fig. 1.18 shows the differences in specific resistance under different environments. It can be seen that the proposed upper loop controller is more energy efficient than the lower loop controller. Over time, the quadruped robot adapts to the environment because the robot maintains a uniform specific resistance as shown in Fig. 1.18. From Fig. 1.19, it can be seen that the body rotation angle is smaller when the loop is crossed than when the loop is not crossed. This is because the intermittent stimulation causes the joint angles to respond instantly to external

sensory feedback and as a result the intermittent stimulation stabilizes the body. This means that as the environment becomes more complex, the central control unit no longer plays an important role [23].

Where v is the velocity of motion, g is the acceleration of gravity, and l is a characteristic length, such as

$$\boxed{} \quad (1.27)$$

the height of the hip joint from the ground while standing. The Froude number is about twice the ratio of the translational kinetic energy of the system to the gravitational potential energy while standing. Thus, the Froude number reflects the exchange of kinetic energy and gravitational potential energy in the dynamics of leg motion. Similarity of dynamics is possible if the Froude number is the same for systems defined in the same way for the different systems being compared. It has been shown that the transition from gait to fast walking and from fast walking to crawling on all fours occurs at approximately the same Froude number values in different animals [24].

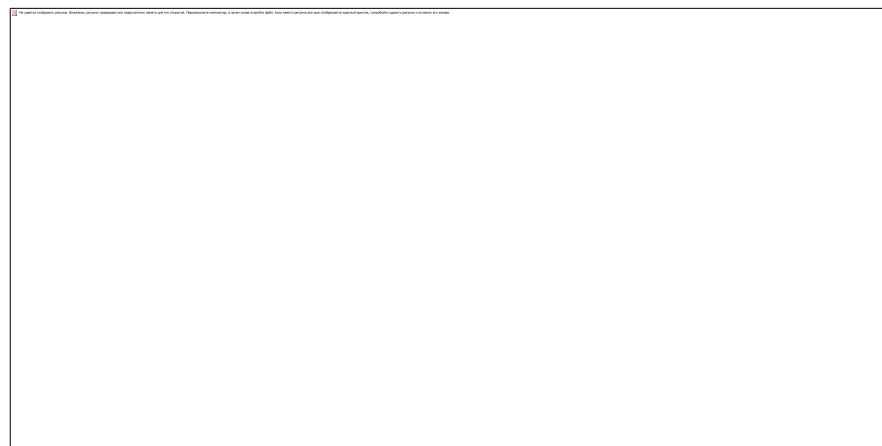


Fig. 1.20 (a) Froudenumber, (b) resistivity

Dynamic simulations were performed on the same model to investigate the relationship between step frequency and Frouden number. As shown in Fig. 1.20, F

and Frouden number change as the step frequency increases. In this case, the energy change decreases rapidly when the step length reaches 1.4 Hz, but there is no abrupt change. However, the simulation results show that $F=1.0$ is lower than $F=1.0$, indicating that animals naturally switch from walking to turning at $F=1.0$ lower than $F=1.0$ [25].

1.5. Walking Pattern

For controlled motion it is necessary to define a gait model consisting of several robot postures; five robot postures are required for a single gait, as shown in Fig. 1.21. The robot legs change one by one to maintain balance, paying attention to the CoM point (center of mass).

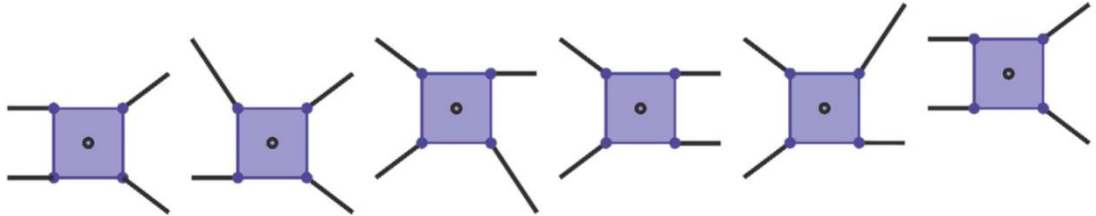


Fig. 1.21 Pose of walking pattern in one periode.

Fig. 1.22 shows the gradual transition from one posture to another in the first half of the gait. The squares on the toe indicate that the legs are on the ground. The dashed line indicates that the leg is raised. Fig. 1.23 then shows the gradual transition from one pose to another in the second half of the gait [26].

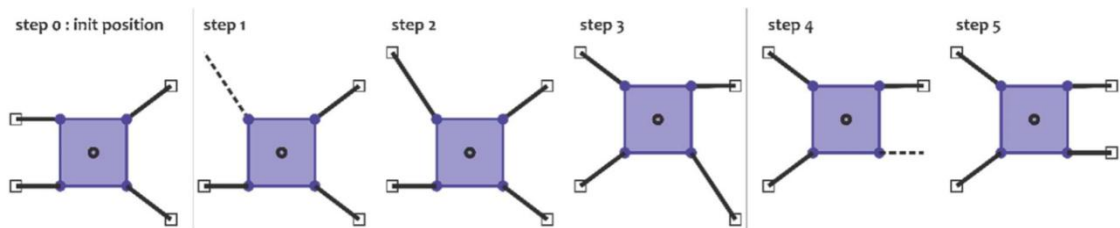


Fig. 1.22. First half periode.



Fig. 1.23. Second half periode

The balancing criterion is to keep the CoM (center of gravity) robot position always within the SP (support polygon); the CoM robot position can be calculated using Equation 11-13.

$$x_{com} = \frac{\sum_{i=1}^n (x_i mass_i)}{mass_{total}} \quad (1.28)$$

$$y_{com} = \frac{\sum_{i=1}^n (y_i mass_i)}{mass_{total}} \quad (1.29)$$

$$z_{com} = \frac{\sum_{i=1}^n (z_i mass_i)}{mass_{total}} \quad (1.30)$$

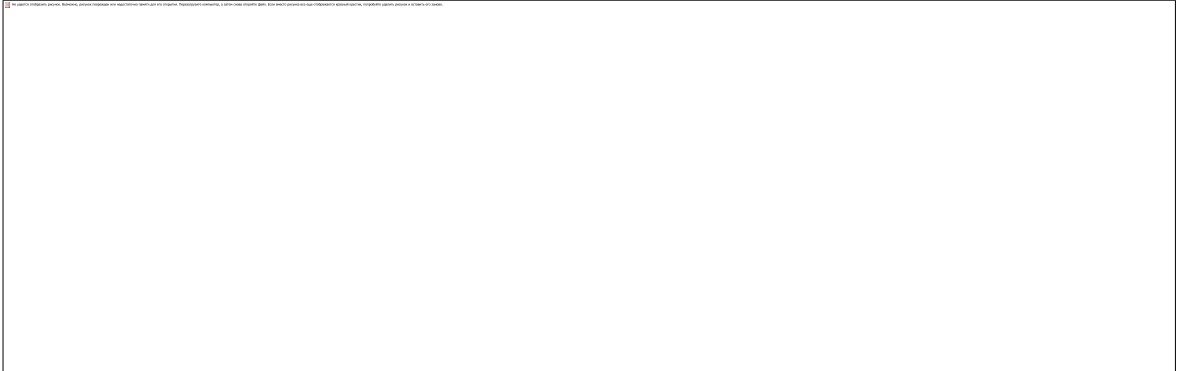


Fig. 1.24. Center of mass point.

The CoM is then projected onto the ground and the value of the Z axis is set to zero. If the position of the CoM is within the range of the support polygon, the robot is in a balanced state. Fig. 6 shows the position of the CoM when four toes press on the floor. If there are only three toes, the CoM's position has moved from its original position to the lower left, but it is still within the support polygon area [27].

1.6. Experimental Result

The experiment used a water-walking robot called AiDIN (Artificial Digitrade for Natural Environment), shown in Fig. 1.1; AiDIN carries most of its components except for power, which is provided by a tether cable.

A Quadruped Walking Robot: AIdin.

As shown in Fig. 1.1, AiDIN has four legs, each with 3 DOF active joints. Each active joint is driven by three geared DC motors (20 W, gear ratio 53:1). In the case of the forefoot, a scapular joint is fitted to change the width of the supporting leg as described above. This is used to switch the ankle movement between passive and active modes. The leg springs, connected between the foot and the shin, are used as elements to store energy and reduce shock during contact with the ground. Two embedded controllers (single board computer, Pentium-III 800 MHz, Compact Flash disk) and 12 microcontrollers (PIC18f458, MicroChip) are installed in the robot. The function of the controller is to control the walking posture as well as to manage the communication between the microcontrollers based on CAN (Controller Network). Another embedded controller (Pentium M. 1.1 GHz, wireless network device). The robot is operated via a cable, but all other communication is transmitted via wireless LAN (local area network) [28].

Experimental Procedures and Results

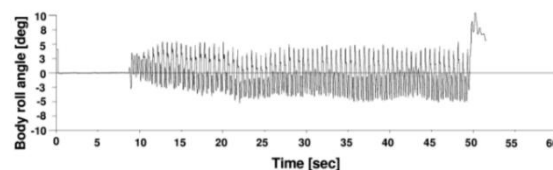
The motion controller was validated through experiments with AiDIN. The first experiment tested AiDIN's ability to change its own posture. In this experiment, AIDIN was placed on a rolling plate as shown in Fig. 1.25 (a) and the plate was suddenly shaken laterally. AIDIN's GLC generated flexion and extension stimuli and stabilized the posture as shown in Fig. 1.25 (b). In the second experiment, the seeded and unseeded loops in the proposed controller were compared, as shown in Fig. 1.25 In this experiment, the robot walked at a walking speed of 20 cm/sec. As shown in Fig. 1.26 (a), the robot rolled at a larger roll angle and sometimes stopped

while rolling. As shown in Fig. 1.26 (b), when two control loops were fully incorporated, the robot was able to maintain its balance even at a smaller roll angle. Thus, it is shown that the proposed control system can control the robot stably and efficiently [29].

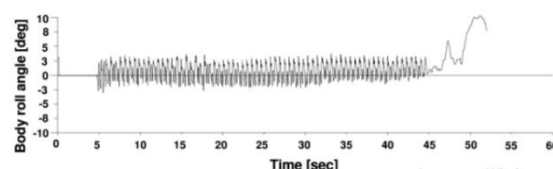
In the second experiment, the seeded and unseeded loops in the proposed control system were compared, as shown in Fig. 1.25. In this experiment, the robot walked at a walking speed of 20 cm/sec. As shown in Fig. 1.26 (a), the robot rolled while expanding the roll angle and sometimes stopped while rolling. As shown in Fig. 1.26 (b), when two control loops were fully incorporated, the robot was able to maintain its balance even at a small roll angle. Thus, it can be seen that the robot can be controlled stably and efficiently using the proposed control.



Fig. 1.25 Experiments to verify self-posture changeability



(a) Trot gait with nonfeedback



(b) Trot gait with feedback

Fig. 1.26 Body roll angle

This section discusses the experimental results of the gait model using UdieBot. This section discusses the experimental results of the gait model by analyzing the robot's position displacement and walking speed. Table 1.11 shows the target coordinates, measured coordinates and error displacements. From Table 1.11, it can be seen that the error increases as the distance displacement increases.

Table 1.11. Displacement error X and Y axis

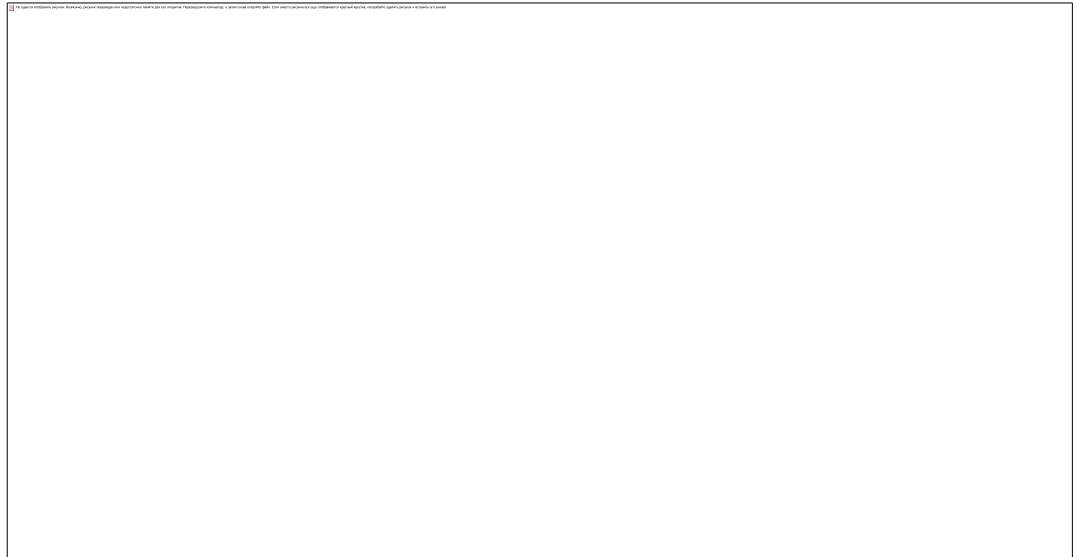


Table 1.12. Speed and Displacement.



The following experiment compares the displacement error at different walking speeds. Table 1.12 shows the robot walking a distance of 100 cm at 10

different speeds. The results show that increasing the walking speed increases the displacement error in the X and Y axes.

CHAPTER II.

WALKING ALGORITHM OF A FOUR-LEGGED ROBOT

2.1. A Four Legged Walking Robot with Obstacle Overcoming Capabilities

Problem Definition

The following studies were carried out in the development of legged mobile robots. Investigating the movement and biomechanics of legged animals and insects, creating models to represent the movement of legged animals and insects, creating mathematical models of balance in legged animals and insects. Creation of models to represent the movement of legged animals and insects. Creation of mathematical balance models for legged robots, simulation of models for legged animals and insects. Designing and building robots, legs and devices for specialized applications; studying animal locomotion and validating locomotion models; designing and building energy-efficient actuators and mobile devices Fabrication. Design and fabrication of energy-efficient actuators and mobile robot legs [30].

This research is concerned with strategies to overcome or avoid obstacles using only information from sensors placed in the feet. Only information from sensors placed on the feet is used to recognize obstacles and manage autonomous sequences of complex integrated movements. Manage autonomous sequences of complex integrated movements to overcome or avoid obstacles while maintaining the robot's balance and stability.

Regarding the design and construction of robots, legs and devices, the following are presented Mechatronic solutions for the construction of anthropomorphic-like robot legs, design of the quadruped robot Guar for statically stable walking, implementation of algorithms for obstacle detection and obstacle crossing. implementation, implementation of kinematics and balance models, implementation of interfaces to generate a static gait, and implementation of all

these functionalities in data structures, interfaces and models, software. Realization of all these functions in a software platform with data structures, interfaces and models built with Object Oriented Programming (OOP). Implementation of all these functions in software platforms with data structures, interfaces and models built with Object Oriented Programming (OOP) [31].

The problems with overcoming obstacles are as follows. Only information from sensors connected to the robot's legs is used. The following. The obstacles considered are step up, step down, step close, and waterway slopes. This approach can be extended to other complex autonomous movements such as stair climbing and other leg-based functions.

The Guará Robot

The Guará quadruped robot shown in Fig. 2.1 is designed as a research robot platform. It is designed as a research robot platform. To walk, detect and overcome known obstacles.

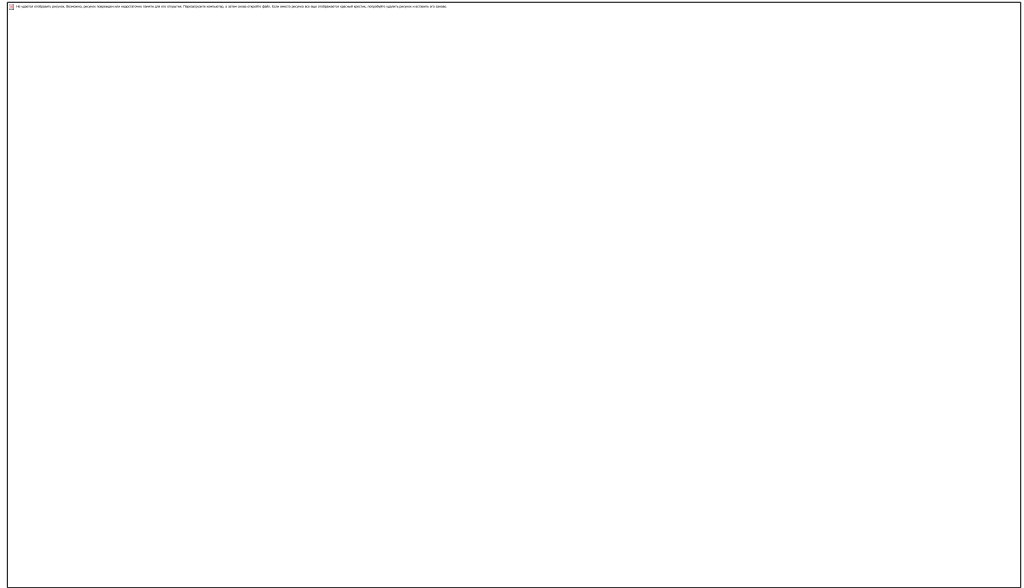


Fig. 2.1 Legged robot Guará turning left while walking.

Moving on Natural Terrain

The robot was tested on natural terrain (meadow with different slopes) and on a flat surface (carpet covering). Control of the movement of the robot was carried out using the proposed neural network motion generator. The parameters of the CPG model optimized for template generation are listed in Table 2.1.

Table 2.1 Optimized parameters of CPG



Examples of pictures of the robot's work are shown in Fig. 2.2 a,b. STIM was set to zero and gradually increased along the time step value ($S_{STIM} = \text{time step}/1200$). The result is shown in Fig. 2.3 a: The CPG model can generate dynamic walking patterns. The robot can successfully generate dynamic walking patterns such as walking, jumping, sprinting, and running on both types of terrain [32].



Fig. 2.2 The robot's dynamic gait pattern (A) on natural terrain (B) on flat terrain. (C) Dynamic gait pattern on flat terrain with injured forelimb. (D) On flat terrain with injured hindlimb. (E) On natural terrain with injured forelimb. (F) On natural terrain with injured hindlimb.

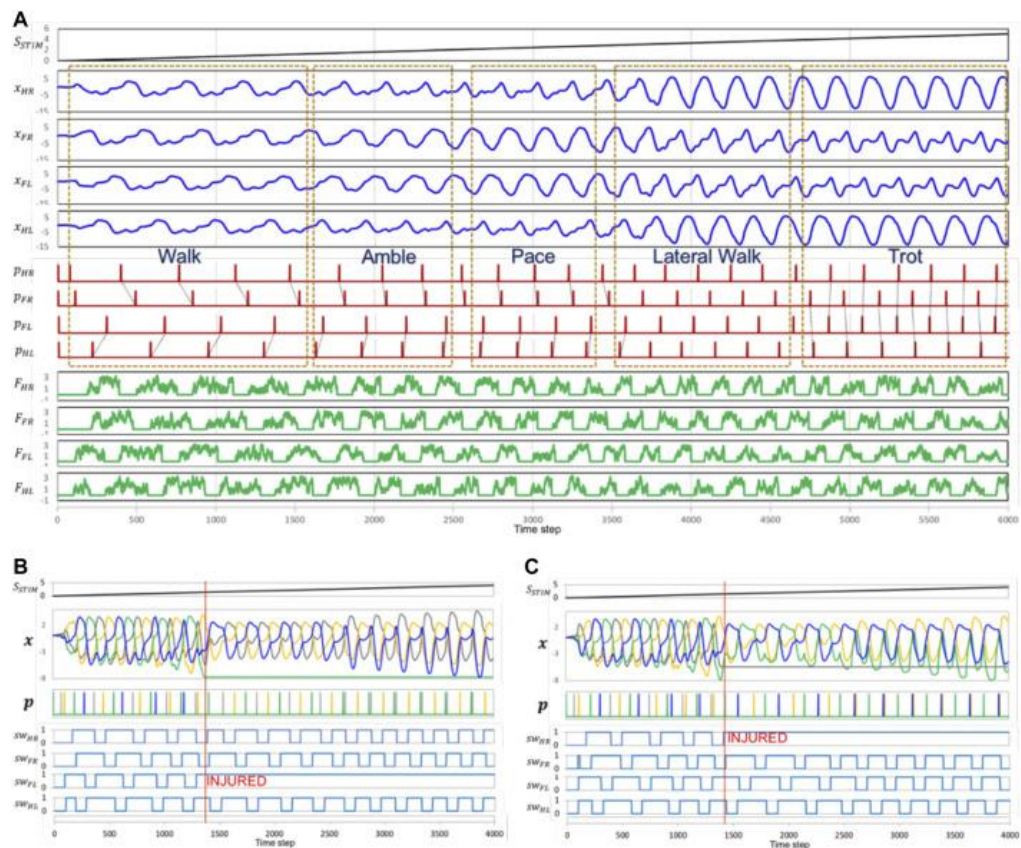


Fig. 2.3 (a) Locomotion models can generate dynamic gait patterns with stimuli of different velocities. Five different known gait patterns from slow to fast can be generated by the CPG model using different speed s STIM stimuli to form a dynamic gait pattern. This increases the frequency of the CPG output signal x for five different known gait patterns from slow to fast. The parameters F_{HR} , F_{FR} , F_{FL} and F_{HL} represent the ground reaction force of each limb. (B) Gait pattern and response to a speed stimulus generated in the defective condition. The p signal model was modified to account for the absence of the CPG signal. The time phase decreases with leg damage. Failure of the right front limb ($N_{FL} = 1$) occurs at 1380 time steps. During injury, the model tends to generate the same phase difference pattern at low velocity. At high speed, the left and right hindlimbs are in the same phase. (c) Failure of the left hindlimb at time step 1400. In this case, the left and right forelimbs are in the same phase at high speed.

Adaptation that depends on the environment

The advantage of using legs for mobile locomotion in animals and machines is the ability to achieve high mobility in harsh conditions by manipulating the contact position of the legs. However, the ability to move in legged robots is still far from the level of animals. Leg movement during locomotion consists of a stance phase, when the legs are pressed against the ground, and a swing phase, when the legs are lifted off the ground. In the stance phase, the legs support the body against gravity, and the interaction between the legs and the ground creates repulsive and braking forces to move the body. The geometric characteristics of the soil can be different. These include flat, sloping, hilly and uneven terrain. The physical properties of the soil also vary. These include hard and slippery soils such as rock, soft soils such as loose soil, and fluid and permeable soils such as sand. The interaction between the foot and the ground is crucial for the execution of movements and requires adaptation of locomotor behavior to ground conditions in real time. Indeed, animals exhibit adaptive coordination between limbs in response to environmental conditions. It is very important to discover and apply the dynamic principles of animals to control legged robots [33].

Manoonpong and colleagues developed a series of CPG-based modular neuromotor control systems for legged robots (Manoonpong et al. 2018, 2019; Steingrube et al. 2019; Goldschmidt et al. 2018; Xiong et al. 2017, 2018; Dasgupta et al. 2020 ; Grinke et al. 2020). They showed that using this control approach leads to adaptive coordination between the limbs, which allows the robot to cope with difficult conditions, such as walking on difficult terrain (Steingrube et al., 2010; Manoonpong et al., 2019; Goldschmidt et al. , 2021; Xiong et al., 2021, 2020; Dasgupta et al., 2019) and obstacle avoidance in unknown complex regions, as observed in insects (Manoonpong et al., 2020; Grinke et al., 2019). For example, they applied modular neural control with adaptive chaotic networks based on CPGs and sensory feedback in a hexapod robot. The chaotic and random dynamics of CPGs are similar to some biological CPGs (Rabinovich and Abarbanel, 2020), and

these dynamics were used to generate different gait patterns depending on environmental conditions. The robot exhibited a quadrupedal gait for standard walking, an undulating gait for uphill walking, a mixture of undulating and quadrupedal gait for downhill walking, and a tripod gait for fast walking. for rapid movement of the optic axis. However, the implementation of such discrete moves does not correspond to the situation observed in insects. In addition to such repeated walking, the chaotic dynamics are particularly favorable for self-recovery of legs from holes in the ground. which leads to the development of support seeking behavior. (2019), Goldschmidt et al. (2020) integrated the improved model into a modular neural control, which allowed the robot to effectively predict the state of walking, as well as extend and lift the legs during the swing and stance phases of cross-country walking. This setup allowed the robot to walk on all fours over rough terrain, climb tall obstacles, and climb stairs with an undulating gait. He also successfully traverses long distances by moving like a caterpillar, moving his left and right legs at the same time. However, in these cases the behavior is more complex than in the case of a caterpillar; Xiong et al. (2020) extended modular neural control by adding a muscle model based on a virtual agonist-antagonist mechanism (VAAM) and created neuromechanical control to achieve leg flexibility. The combination of neuromechanical control and sensorimotor learning enables energy-efficient walking in different types of gait with appropriate leg flexibility (Xiong et al. The robot walked efficiently on various surfaces, including sponge, gravel, fine gravel, and grass. To adapt to obstacle avoidance in challenging environments, it was implemented an adaptive neurosensory computing network for modular neural control (Grinke et al.) An adaptive computing network can control various turning methods using the robot's short-term memory As a result, the robot has evolved and adapted its turning behavior to avoid obstacles, sharp corners and dead ends in different situations. In addition to modular neural control [34].

Horses decelerate from small to large strides when climbing slopes (Wickler et al., 2018) or when carrying loads (Farley and Taylor, 2017). Hexapods, such as

bats, cockroaches, and beetles, change their gait depending on the slope of the ground (Spirito and Mushrush, 2017; Pelletier and Caissie, 2018; Grabowska et al. In addition, cockroaches use a tripod gait during normal walking, but when they are attached to the ball, their gait changes to a metachronous gait (Spirito and Mushrush, 2021) have a similar effect on the gait of cockroaches (Fujiki et al. (2019a). work on a six-legged robot and showed that load- and angle-dependent sensorimotor interactions, as observed in insects, can alter gait between tripedal and semipedal gait.

Fukuoka et al. (2021), Fukuoka and Kimura (2020) and Kimura et al. (2020 a,b) controlled a bipedal robot (Tekken series) using neural oscillators developed by Matsuoka (2017). They included models of various reflexes, such as flexion reflex, extension reflex, and vestibulospinal reflex, based on sensory information. They also modeled the tonic responses of the maze to coordinate rolling movements, which must be synchronized with bending movements. The robot reliably moved over uneven terrain, including steps and slopes, facilitating the transition from walking to crawling [35].

When the soil is liquid, for example, sand, the feet penetrate deeper into the ground during movement. This leads to complex interactions with the soil that create lift, drag, and thrust [see a review by Aguilar et al. (2019). Lee et al. (2019, 2021) used the gait of a three-legged robot for a six-legged robot and applied a force model of a robot moving in a granular environment to adjust the leg shape and leg movements. To create movement similar to movement on solid ground.

2.2. Leg Design and Reference System

There are six movements in a normal human stride. These movements straighten the circular path of the hips by increasing the length of the step and the effective length of the supporting leg. They smooth the transition between the two leg supporting strides. The two legs soften the transition between the supporting step. Transfers weight to the supporting leg during the leg transition.

Guara movements are inspired by normal human steps. Guara. leg has two rotational joints that connect the femur and hip joint, which perform pitching and rolling. There are two pivot joints that connect the femur to the hip joint and perform the pitching and rolling of the femur. The femur is connected to the tibia, which performs the pitching and rolling of the knee, the tibia is connected to the foot and performs the step of the foot. The 16 leg joints are driven by a toothed belt driven by a DC motor and reduction gear mounted on the robot body. They are driven by a reduction gearbox. This joint arrangement enables allowing lateral movement of the hip joints increases the stability of the robot by allowing the robot's center of gravity to move towards the support polygon. By moving the center of gravity towards the support polygon, the stability of the robot can be increased [36].

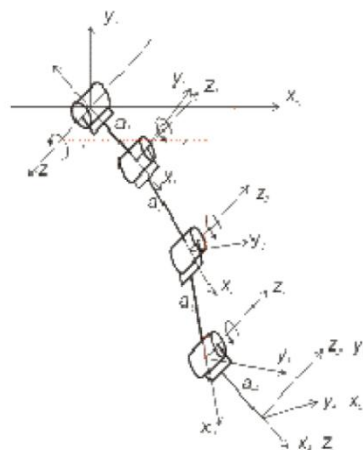


Fig. 2.4: Leg revolute joints and joint reference systems according to Denavit-Hartenberg

2.3. Leg's Inverse Kinematics

A typical method to obtain the inverse kinematics of a manipulator with six degrees of freedom is to obtain the reference position of the wrist from the desired position and orientation of the end effector and use geometric methods for the first three joints and then the three wrist joints.

Legs with four degrees of freedom are an unusual problem in inverse kinematics because there is no complete rotation matrix for the end effector, so this method cannot be used. In this case, however, since the last three joints are all parallel and perpendicular to the first joint, it is possible to solve the inverse kinematics geometrically directly from the joints. This scheme was used by Bonitz (1997) to find the inverse kinematics of the 4-DOF manipulator used on the Mars Volatiles and Climate Surveyor (MVACS) launched by NASA [37].

Looking at Fig. 2.2 and applying the cosine law to the triangle formed by joints 1, 2 and 3, we obtain

$$\cos \theta_3 = \frac{x_3^2 + y_3^2 + z_3^2 - (a_2^2 + a_3^2)}{2 * a_2 * a_3} \quad (2.3.1)$$

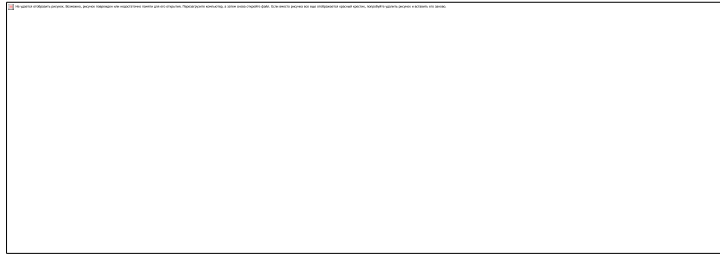
Geometrically, θ_2 can be obtained from Fig. 1.2 as follows:

$$\theta_2 = \tan^{-1} \left(\frac{z_3}{\sqrt{x_3^2 + y_3^2}} \right) - \tan^{-1} \left(\frac{a_3 * \sin \theta_3}{a_2 + a_3 * \cos \theta_3} \right) \quad (2.3.2)$$

From the coordinates (x_3, y_3, z_3) of the shaft end in Fig. 2, the rotation θ_1 is obtained as follows:

$$\theta_1 = \tan^{-1} \left(\frac{x_3}{y_3} \right) \quad (2.3.3)$$

The foot moves with θ_4 always known, which gives the coordinate of the joint 3 in relation to the supporting point coordinates (x_4, y_4, z_4) as follows:



(2.3.4)

The mapping of foot coordinates from motion space to joint space is as follows. The mapping is done with equations (1) to (4). The matrix of the realized walk is as follows. The trajectory of each foot in the working space of the leg is synchronized with the lateral movements of the body and mapped into joint space by the inverse kinematics of the leg. The inverse kinematics of the legs map them into joint space [38].

2.4. Foot Stroke and Flight Trajectories

The trajectory of the foot strike is a straight line. In. Other configurations are possible, such as walking on an inclined path while the robot body is kept horizontal, or walking on a horizontal path while the robot body is inclined. The robot walks on a horizontal path with the robot body inclined.

Any continuous foot flight trajectory can be applied to ensure that one foot does not hit the other foot and make contact with the ground. During the flight to a new striking position, the other foot must not collide with the ground.

The ideal foot flight trajectory is ideally curved with the boundary condition that the foot has zero speed during take-off and landing. The ideal foot flight trajectory has the boundary condition that the velocity is zero during take-off and landing, and the ideal foot flight curve is a cycloid. However, due to the flexibility of the legs, the mechanical propulsion of the joints and the stability of the robot, the legs stayed on the ground a little later to the new strike position at the beginning of the flight. The legs were a little late in reaching the ground. At the end of the flight. At the end of the flight, a little before the desired landing position, the legs are always earlier than the cycloid model predicts. During these periods. During these periods, the feet are moved in the opposite direction of motion, causes reverse thrust during take-off and landing, leading to unwanted balance and instability in the robot and causes the robot to become unstable [39].

To avoid this, a polygonal trajectory was created, as shown in Fig. 2.5, with leg thrust from right to left, vertical distances measured from the coordinate system origin of the leg and dimensions in millimeters.

At the take-off point associated with the left inclined section (blue) of the generated trajectory, the leg moves upwards with a horizontal velocity equal to the robot velocity and a higher vertical velocity until the leg is completely off the

ground. It then accelerates to reach a higher horizontal velocity, defines a trajectory consisting of three upward segments and, at the end of the third upward segment, initiates a descent motion on the right inclined segment of the generated path. During the descent, the horizontal speed is adjusted to the robot's travel speed and the vertical speed is reduced until the feet touch the ground. The receiving line (red) is the foot path generated by direct kinematics of the joint coordinates received from the robot. Note that due to the mechanical collision of the joints, the foot take-off starts at the predicted time which deviates from the transmitted values, and there is a reverse stroke in the landing and take-off of the foot [40].

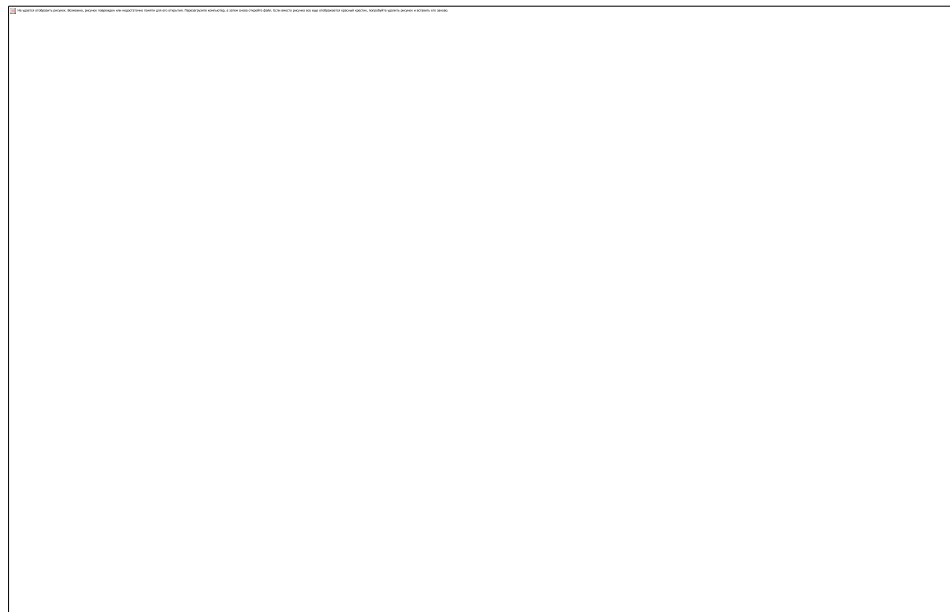


Fig. 2.5: Leg 3 joint 3: generated and received paths in operational space, saggital plane

Robot Design and Reference Systems

Fig. 2.6 shows the robot flying leg 3 to a new stroke position and starting to walk. The legs of the robot are numbered as follows: 0 front left leg, 1 back left leg, 2 front right leg and 3 back right leg. The main dimensions are as follows: l : length, w : width, g : geometric center, G_{xy} : projection of the geometric center on the ground.

The global reference system $({}^0x^0, {}^0y^0, {}^0z^0)$ and the robot reference system $({}^r x, {}^r y, {}^r z)$ are parallel when the robot starts walking. Regarding the robot reference system, each leg has its own reference system, as shown $({}^0 x, {}^0 y, {}^0 z)$ for the leg 0 in Fig. 2.6.

Robot Kinematic Model

As shown in Fig. 2.7 (a), (b), (c) and (d), there are four possible combinations of two leg position vectors ${}^r P_{ij}$. Where the right superscript r is the on-board robot reference frame and the left subscript ij is the leg number.



Fig. 2.6: Joints and robot reference systems

For each support pattern, the cross product of the unit normal vectors is calculated, allowing a rotation matrix to be defined between two consecutive set points in the on-board reference system. As the robot walks, the rotation matrix is updated and the final rotation from the starting position is calculated step by step, which can be checked and corrected by the on-board compass and inclinometer or by the stereo orientation reference of other robots [41].

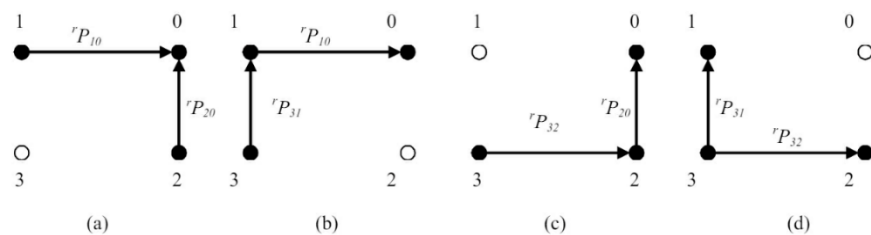


Fig. 2.7: Legs supporting patterns.

CHAPTER III

STATICALLY STABLE GAITS

3.1. Curved Paths

With a statically stable gait, the robot's leg must stand on the ground with at least one wooden leg. Fig. 3.1 shows a periodic positive symmetric walk. The walking load factor β is the ratio of step time to total walking time, and it is the same for all legs. The leg phase angle ϕ is the ratio of the landing time to the total walking time.

A gait matrix describes a gait pattern with rows associated with legs and columns associated with leg states. In a walk matrix, β is the ratio between the rows with single entries and the sum of the columns, and ϕ is the ratio between the row of the column containing the first single entry after the zero entries from left to right and the sum of the columns.

In order to find the ideal values of β and ϕ , a gait matrix generation library is also available in the robot software. The gait matrix is directly entered into the gait generation library, which determines the gait load factor β and the phase angle ϕ of the legs. The total number of columns defines the gait cycle, the total number of entries 0 defines the flight part of the gait cycle, and the 1 entry - the step part. The gait matrix and all its properties can be tested using the robot GUI and/or robot gait. The 32-segment gait in Fig. 6 has 27 steps and 5 flight segments with a load factor of $\beta = 0.84375$ [42].

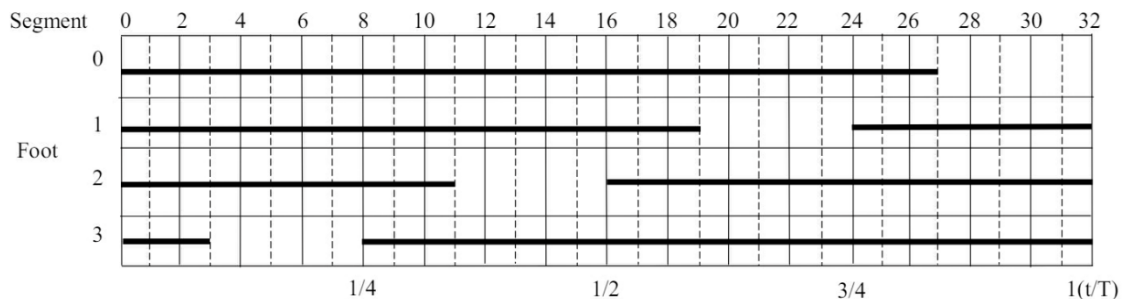


Fig. 3.1 Regular symmetric gait.

In Fig. 3.1, the gait cycle begins at segment 0, where the four legs perform movements to move the robot body forward and to the left relative to the legs. In segment 3, back right leg 3 flies into a new kicking position, and in segment 8, all four legs kick again. In segment 11, front leg 2 on the same side flies to a new kick position, and in segment 16, four legs perform a kick to move the robot forward and to the right, and then the sequence of movements is repeated for rear left leg 1 and front left leg 0 [43].

The coordinated movements of the legs that move the robot are synchronized by a gait matrix and an addressing algorithm that moves the robot's body laterally during a straight or curved gait trajectory.

Curved Paths

Fig. 3.2 shows a descent along a curved trajectory. The implemented algorithm is a controllability strategy that allows the work to be turned while walking without losing balance and stability. The maneuverability strategy allows the work to be turned while walking without losing balance and stability. Curved tracks are incremental quadropods that are tangent to the robot's curve and are inclined with respect to its longitudinal direction.

The differential stroke is tangent to the curve of the robot and inclined in relation to its longitudinal axis. The movement trajectory is realized by small linear segments in the working area of the legs. Here θ is the angle of rotation, R_o and R_i radii of the outer and inner link of the curve, λ_o and λ_i - step of the outer and inner link of the curve, L and λ_i - the intervals between the strokes of the outer and inner links of the curve, L and W - the length and width reference systems between the links System [44].

$$\lambda_o = \theta * R_o = \theta \left[\left(R_o + \frac{W}{2} \right)^2 + \left(\frac{L}{2} \right)^2 \right]^{1/2}$$

$$\lambda_i = \theta * R_i = \theta \left[\left(R_o - \frac{W}{2} \right)^2 + \left(\frac{L}{2} \right)^2 \right]^{1/2} \quad (3.1.1)$$

Straight trajectories are also applied with small linear steps, the only additional computational effort required to return the robot to the curvilinear path is to sum the four 3x1 matrices with small line segments that form the curvilinear path of the leg in the leg's working area to find the position of the next leg. The four 3x1 matrices are simply summed with small line segments that form a curvilinear foot path in the foot workspace to find the position of the next foot. Inverse kinematics then maps the Cartesian coordinates of the foot to the joint coordinates of the foot [45].

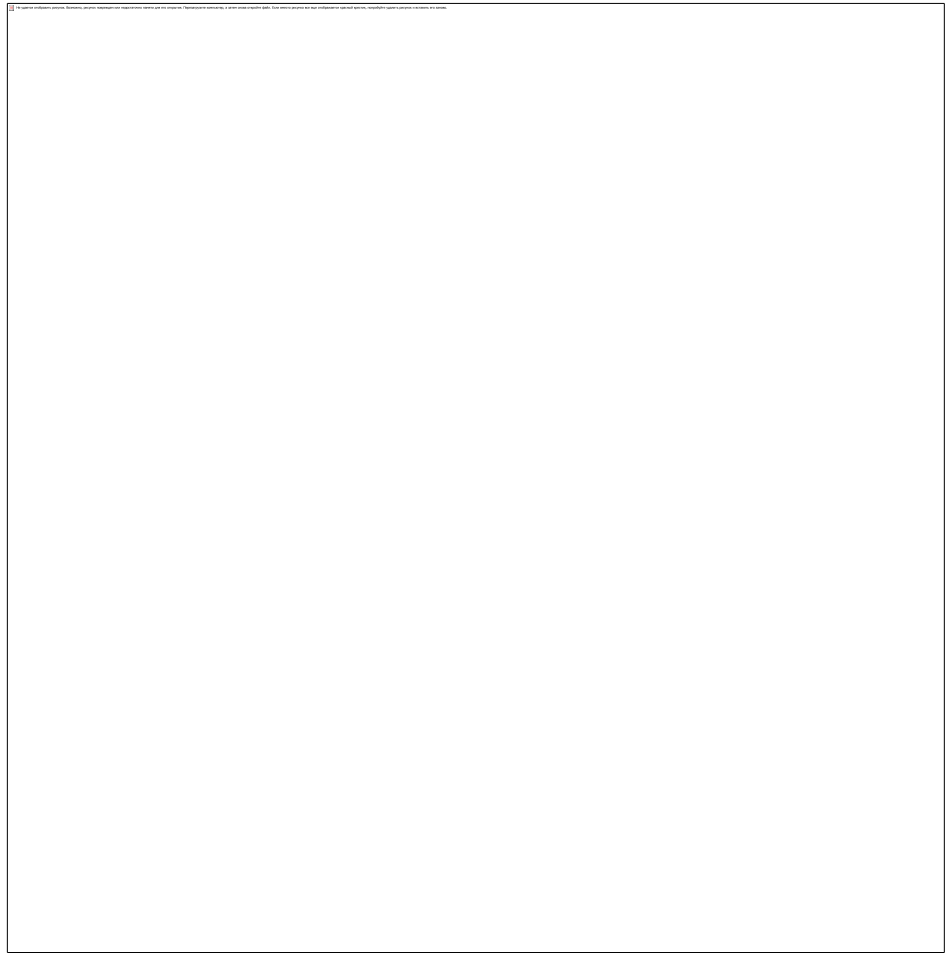


Fig. 3.2 Curved path stroke.

3.2. Obstacles Detection and Overcoming

Information about the environment surrounding the robot is used to plan its path. In addition, specialized recognition systems are widely used to detect low, narrow or aerial obstacles in the robot's path.

This paper proposes an approach to the movement problem of SCARA quadruped robot "JROB-1" and an obstacle detection scheme on the path of robot legs using proximity sensors to improve robot movement in a complex environment.

This work proposes a strategy to detect and overcome obstacles in the trajectory of robot legs using contact switch collision sensors placed on robot legs. Fig. 3.3 shows a contact collision sensor mounted on Guar's legs. Fig. 3.3 shows the contact switch collision sensor mounted on Guara's legs.

There are two pairs of switches mounted horizontally in parallel on the front and rear legs, and two single switches mounted vertically on the front and rear legs. Upon collision, the system identifies an obstacle using a circuit of switches and initiates an integrated movement sequence that either moves the robot or pushes the obstacle aside [46].

Fig. 3.4 shows the location of the switches on the footswitches. Patterns [1011] in Fig. 9(b) and [1101] in Fig. 9(c) are horizontal, patterns [0111] in Fig. 9(a) and [1110] in Fig. 9(d) are vertical, and pattern [1111] in Fig. 9(e) is contactless.

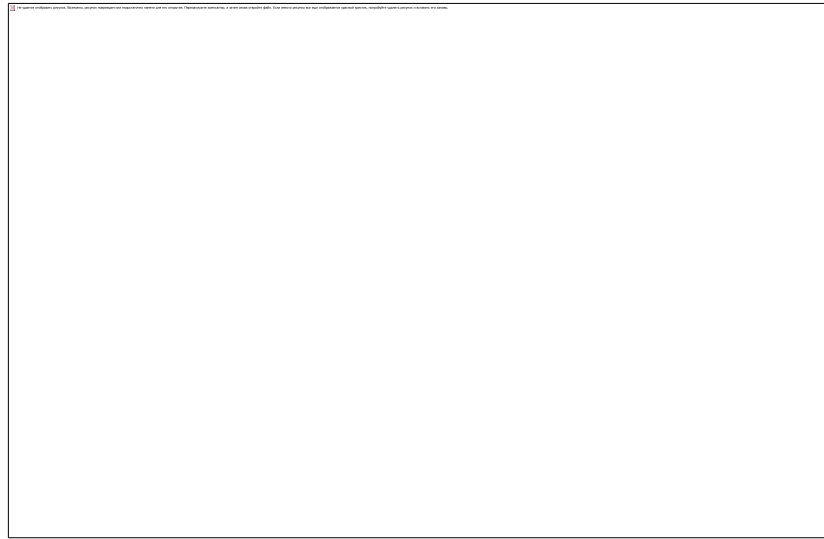


Fig. 3.3 Contact keys on the feet

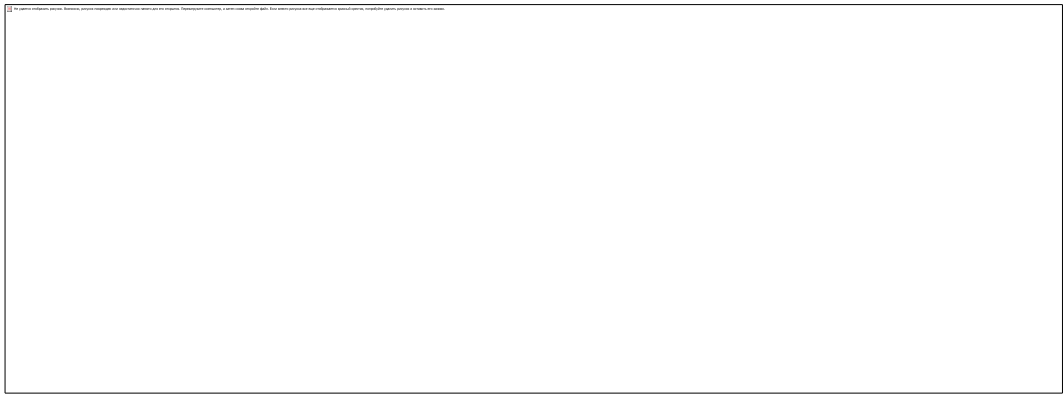


Fig. 3.4 Feet touch contact keys

The switching of contacts also occurs during take-off and landing events before and after the foot flies to a new striking position, or when crossing an obstacle. The motion sequences studied are for step up, step down, channel, step close, ramp and steering obstacles.

The main obstacle clearance control loops are shown in Fig. 3.5 Each obstacle has a unique recognition key model and other obstacle models can also be applied. The step-by-step obstacle clearance control loop is shown in Fig. 3.6.

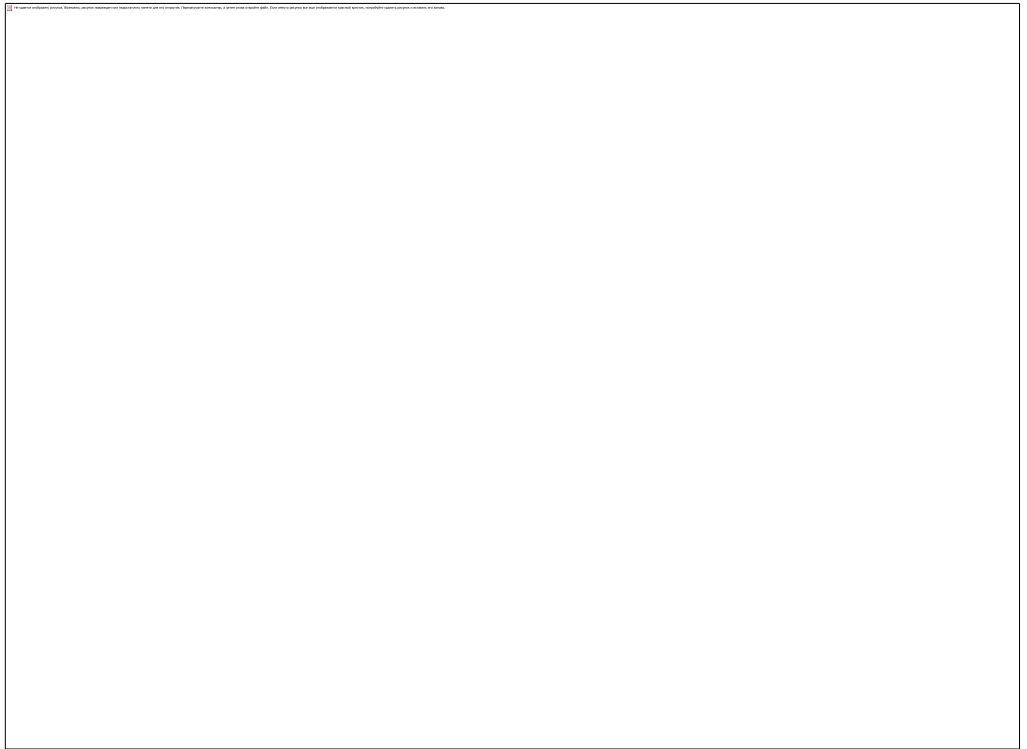


Fig. 3.5 Obstacle overcoming control cycle.

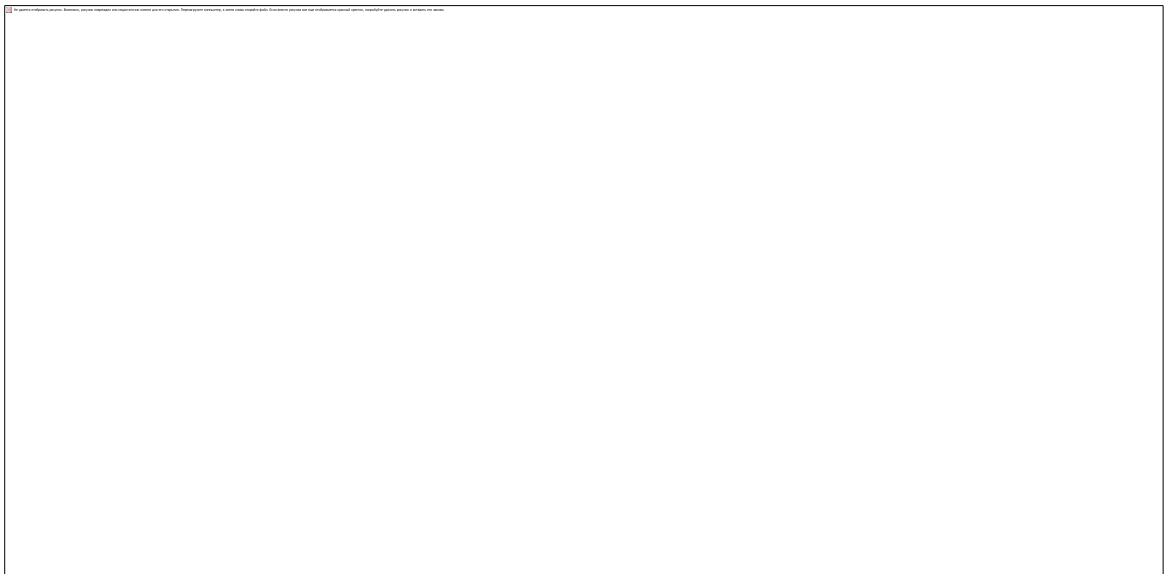


Fig. 3.6 Step up obstacle overcoming control cycle.

3.3. Main Control Structure

The robot's system control architecture is showed in Fig. 3.7 The three layers control system structure comprises a supervisor and a coordinator level connected by a CAN bus and an actuator level. Processes associated with walk and overcoming obstacles always reach states that start other processes which still can start other processes, up to 3 levels. All the main control cycle tasks run in the supervisor level. The robot's main control strategy is based in "case" and "if then else" patterns which sense and react to contact keys state changes; executes the first level task which generates direct kinematics of legs according to the kinematic model, constraints and kinematic limits of the robot; process the inverse kinematics; and communicates with robot on board controller. The second and third state levels run processes which manage groups of movements to perform a task in hand [47].

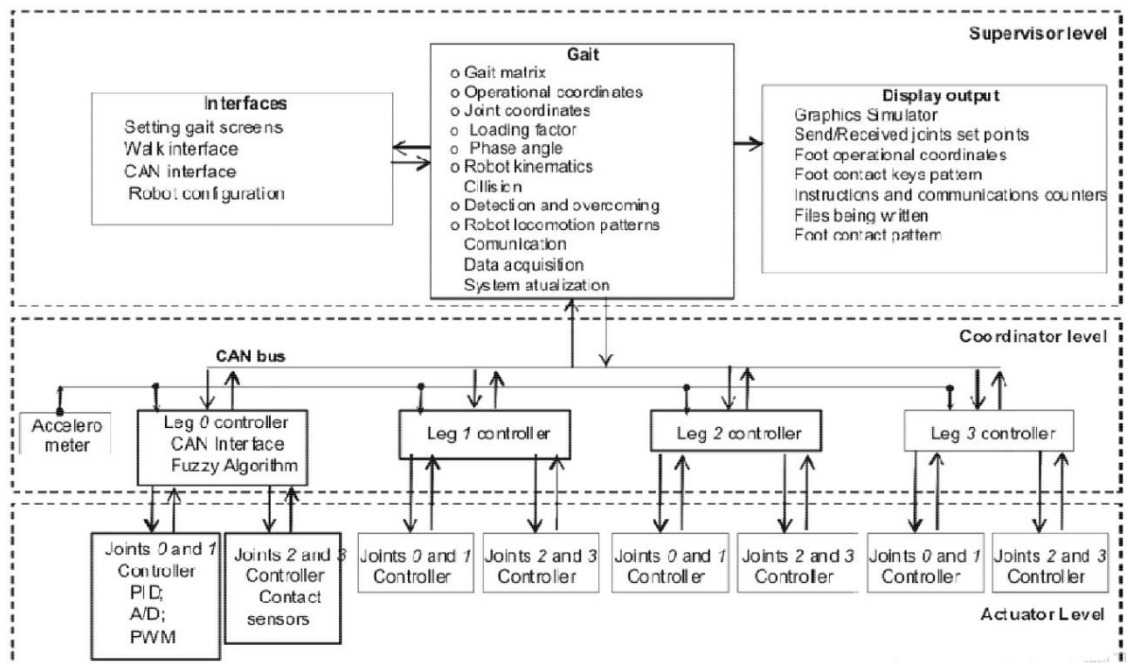


Fig. 3.7 Control system architecture

The main control loop is shown in Fig. 3.8. The internal processes are a collision processing unit that manages the obstacle avoidance task, a gait processing unit that manages the walking task, and a communication unit that manages the hardware communication between the control system and the robot.

In Fig. 3.8, the main unit executes a fixed sequential polling process at the first processing level, checks all processing branches and executes the communication unit process at the end of each control loop.

The gait processing unit performs the gait task to gradually build the robot's straight and curved path according to the conFig.d gait parameters.

The collision processing unit manages the obstacle movement sequence and is triggered by the main unit when the switching pattern switches to a pre-programmed known obstacle. When a collision occurs, the control loop of the main unit activates the collision unit and all operations take place within the collision unit until the robot returns to the walking state.

Operations related to the robot's communication and human-machine interface are performed in the communication unit in each control loop. At the coordinator level, the joint controller and the CAN network bus are realized with the supervisor level. The actuator level realizes the direct control of the joints and the contact switch data acquisition [48].

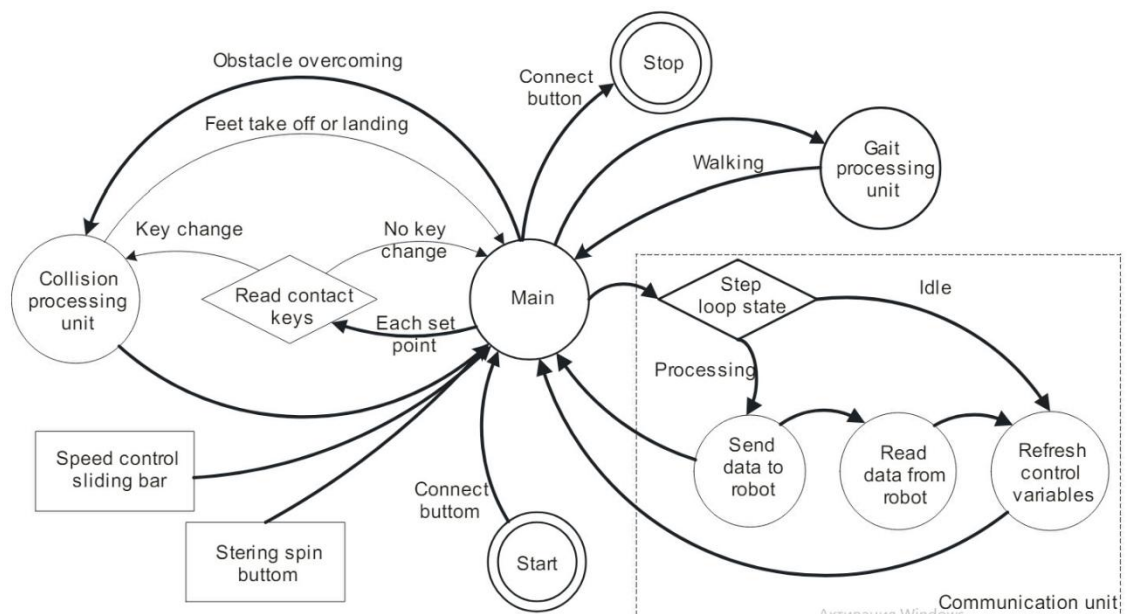


Fig. 3.8 Guara's main control cycle

Results

Guara robot

Fig. 2.1 shows a photograph of the Guara robot. The walking system of this robot. The walking system consists of four anthropomorphic-like legs electrically driven by DC motors and reduction gears. The gait system consists of four anthropomorphic-like legs electrically actuated by DC motors and reduction gears, and the navigation and balance scheme is based on human walking. Obstacles. The robot's ability to overcome obstacles is based solely on information obtained from contact sensors placed on the robot's legs. Only the information from the contact sensors is used. This information is used to manage an integrated sequence of movements to turn or deflect obstacles [49].

Robot control system software

Developed in OOP, Guará software is an integrated environment for robot driving, gait development, parameter tuning and robot diagnostics. Guará software is an integrated environment for robot driving, gait development, parameter tuning and robot diagnostics. The software environment also allows the development of new integrated motion sequences for the transfer of new obstacles by adding new obstacle code libraries in C++.

The robot gait initialization starts with the loading of the selected gait matrix file. The start of the robot's gait starts with the loading of the selected gait matrix file. The remaining strokes are defined until each leg reaches its kinematic limit. Then, the number of points in the workspace to build the trajectory of the legs, the maximum stroke to be used (below the minimum stroke for the legs available) The user then enters the maximum stroke to be used (below the minimum stroke of the current leg). The user then enters the following information .

3.4. Operational and joint coordinate in straight walking

Fig. 3.9 shows the working coordinates (x_3 , y_3 , z_3) of leg 0, joint 3, generated and executed in the reference frame of the robot according to the Denavit-Hatenberg law, using 108 hit points and 20 flight points to generate the foot trajectory.

As can be seen from Fig. 3.1, the basic gait model uses four points for each segment. According to Fig. 3.1, foot 3 starts flying after 3 segments and flies for 12 steps. The robot starts walking by moving its body to the left movement limit. As shown by x_3 in Fig. 3.9, the body is moved to the left locomotor boundary for the flight of leg 3 and leg 2. Next, Fig. 3.1 shows that in segment 16, corresponding to the number of steps 64, the robot moves forward and moves forward and moves the body to the right limit of movement. This is the last foot to fly to the new batting position. Coordinate z_3 is the batting position to the new position, where y_3 represents the flight of the foot in the same step number interval, and Step count interval [50].

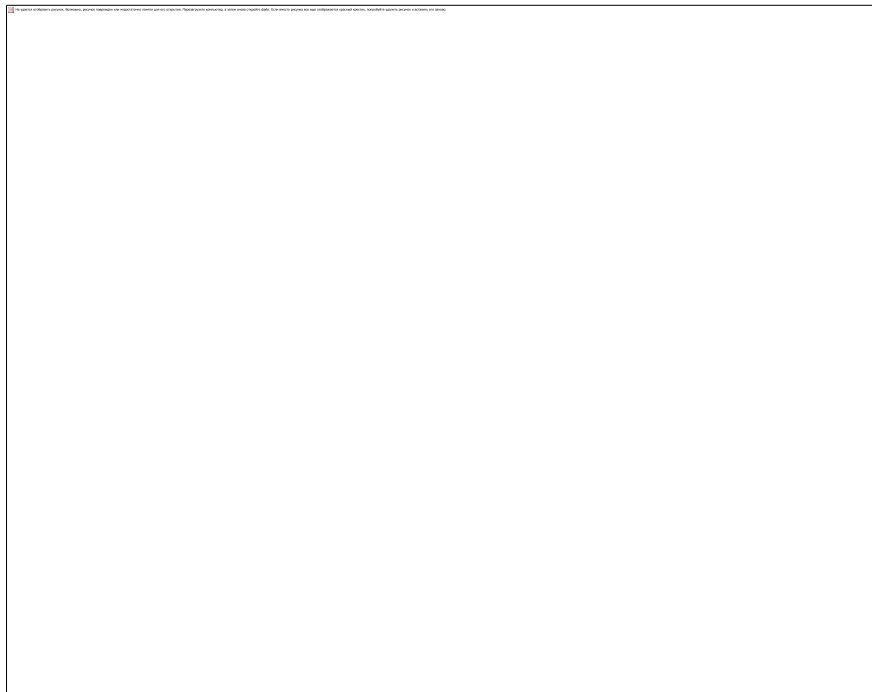


Fig. 3.9 Operational coordinates generated and executed for leg 0, joint 3, robot's reference frame.

It can be seen that the time it takes for the feet to fly to the new position represented by the y_3 coordinate synchronized with the z_3 transverse path is very short. It is represented by the y_3 coordinates and synchronized with the z_3 transverse path. From the representation of the x_3 coordinates, it can be seen that leg 0 flies only at the end of the x_3 transverse motion and after the flight of leg 2. It can be seen that it flies only at the end of the x_3 lateral movement and after the flight of the 2nd leg.

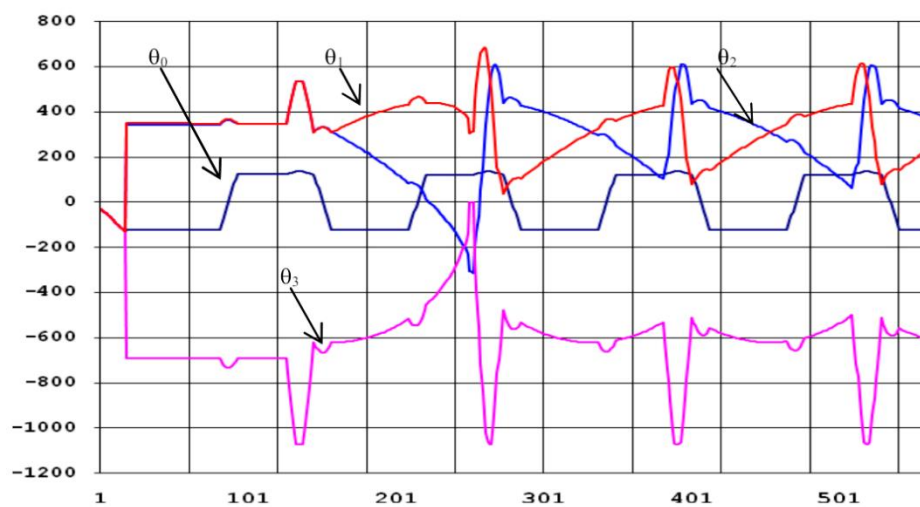


Fig. 3.10 Joint coordinates generated and executed for leg 0, joint 3, robot's reference frame

Fig. 3.10 shows the joint coordinates ($\theta_0, \theta_1, \theta_2, \theta_3$) generated and executed for leg 0 joint 3 in the robot's reference frame using 108 strike points and 20 flight points to generate the foot trajectory.

The angle θ_0 is the rolling degrees of freedom of hip 0, which moves the robot body laterally to the left motion boundary at the beginning of the robot's gait. The θ_0 lines show a straight section where the leg on the same side flies to a new position and a pinched section corresponding to the lateral movement of the robot body to the left and right. The robot body remains in the horizontal plane due to the counterclockwise rolling motion for all legs. A clockwise rolling motion is then

applied to all legs to move the robot body to the correct kinematic boundary. The undulation at the end of each straight segment corresponds to the flight segment of the leg and requires a larger movement of joint 0 to maintain the vertical motion of the leg during flight.

The angle θ_1 is the degree of freedom of hip 0, which moves the thigh in the anteroposterior direction. The discontinuous surface of this curve also occurs in the flight section of the foot where greater joint angle abduction is required. There is also a small fluctuation that synchronizes with the section where θ_0 changes the robot's body position from the left/right motion boundary, which requires an increase in leg length.

The angle θ_2 is the knee joint 1-step degree of freedom that moves the shin back and forth. The same fluctuations are seen as described for θ_1 and this occurs simultaneously with the leg flight range.

The angle θ_3 is the ankle joint 2-step DOF, which rotates the foot, and the same fluctuations are seen as described for θ_1 , and this occurs simultaneously with the foot flight range for the same reasons as above.

3.5. Operational and joint coordinate in curved walking

Fig. 3.11 shows the stroke and flight path of the left forefoot 0 and right forefoot 2, Hit and flight trajectory the radius is about 1.4 [m] and the center of curvature is on the right. It can be seen that the stroke step of leg 0 is larger than that of leg 2. Fig. 3.2. As shown in Fig. 3.2, according to the proposed model, leg 0 has a larger radius because it is outside the curve and needs to be moved to a greater extent with each number of steps.

Fig. 3.12 shows the roll angle θ_0 of the left front leg 0 and the right front leg 2. The curve pattern shows that the slope of the line reverses at the thrust period corresponding to the lateral displacement. In contrast to the straight lines of the thrust straight path, it can be seen that the slope of the lines reverses at the thrust period corresponding to the lateral displacement. This contrasts with the straight lines of the straight thrust path (Fig. 3.9). These lines correspond to feet 0 and 2. They correspond to a lateral reversal motion that rotates the robot body to the right.



Fig. 3.11 Front left foot 0, and front right foot 2 stroke and flight paths in leg's reference frame.

The discontinuity seen in the joint roll angle θ_0 curves of leg 0 and leg 2 is due to the vertical lifting of the leg and this is due to the additional movement required for the legs to be lifted vertically and move to the new striking position while the robot body moves laterally. This is due to the additional small run required for the feet to be lifted vertically and move to the new striking position while the robot body moves laterally.

Fig. 18 shows a Guala robot with a radius of about 90^0 mm. It is shown turning right at a corner. In Fig. 3.13 (a) the front right foot 2 is flying towards a new striking position. In Fig. 3.13 (b) the foot is on the ground and the rear right foot 3 is flying. In Fig. 3.13 (b) the foot is on the ground and the right hind foot 3 is flying. Fig. 3.13 (c). In Fig. 3.13 (c), the robot moves forward and makes a right turn with the front foot striking out from the center and the back foot striking out from the center of the curve. In Fig. 3.13 (c), (d) and (e), the outer legs 0 and 1 move to a new stroke position. Stroke.

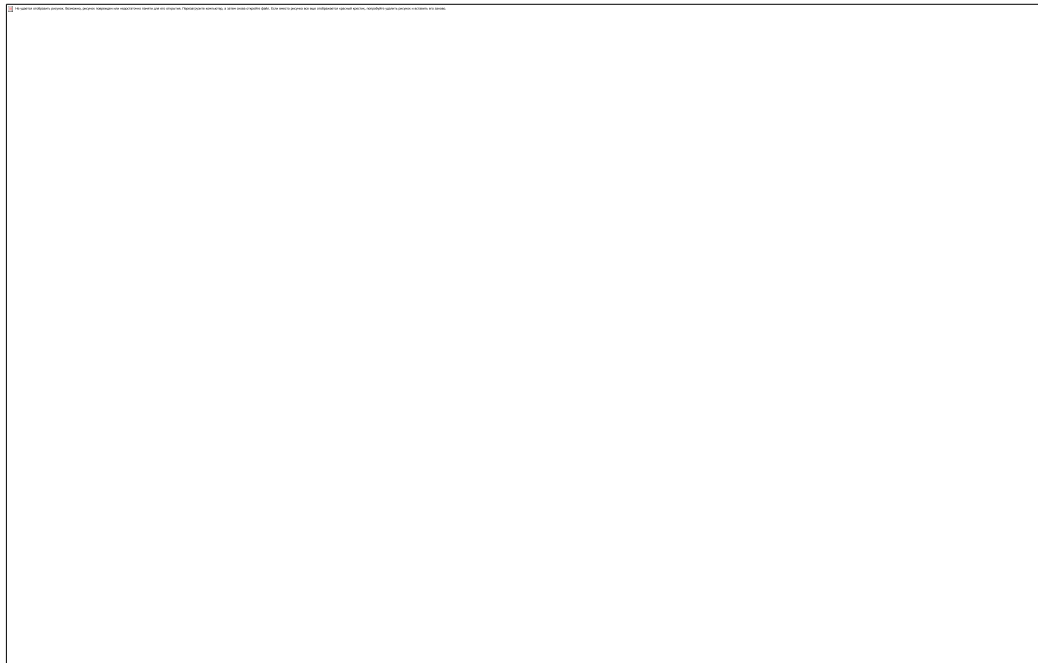


Fig. 3.12 Joint coordinates generated and executed for leg 0, joint 3 in leg's reference frame.

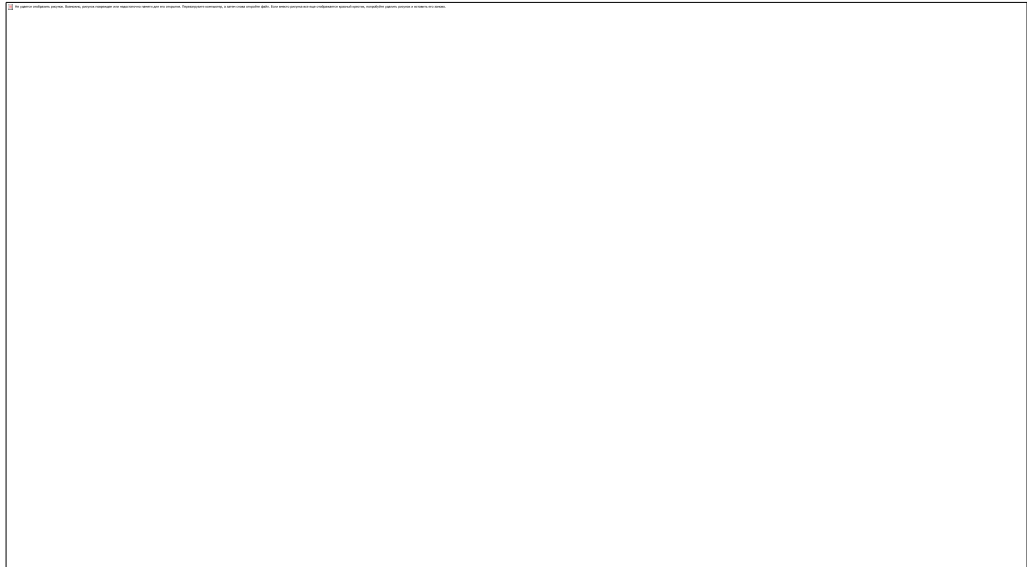


Fig. 3.13 Guar robot turning right in a 90⁰ corner.

Application for overcoming barriers

To overcome obstacles, the recommended approach uses only the foot contact keys. To move the robot to overcome the obstacle, only information is used and move the robot over or through the obstacle while maintaining its balance and stability. To start overcoming obstacles, move the robot as follows before the legs are moved in the air. Move the robot in the following direction move the robot over there.

3.6. Overcoming a step up obstacle

The Guara quadruped robot is shown in Fig. 3.13,

Figs 3.14 and 3.15 show climbing a step in front of the walkway. Steps are infinitely arranged and the robot can only recognize them when one of the keys on the forefoot is touched. In Fig. 3.13 (a) the robot recognizes that the claw key is touched.

In Fig. 3.13 (a) the robot is walking and the right hind leg 3 has made a new stroke in Fig. 3.13 (a) the robot is walking and the right hind leg flies to 3 new stroke positions. Fig. 3.13 In (c), the robot has initiated a motion sequence that takes the right forelimb 2 back to the flight start position and lifts it up. It brings it to the flight start position and moves the robot body left or right up to its kinematic limit, then starts the motion sequence to bring leg 2 to the stepping position.



Fig. 3.14 Front feet going up a step.

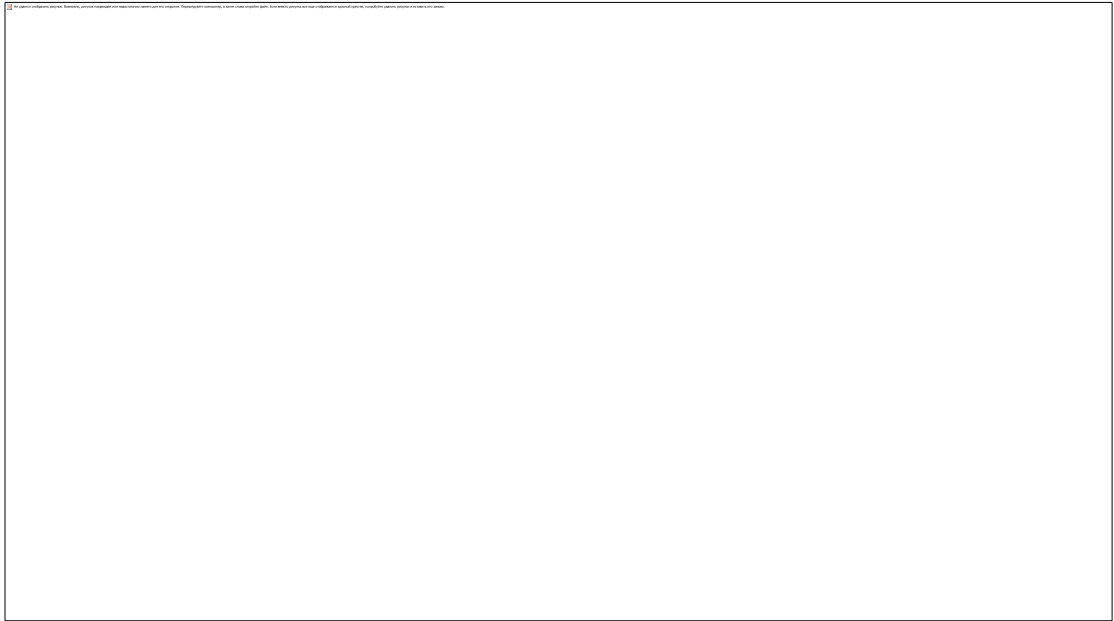


Fig. 3.15 Positioning the rear legs on the step up to restart walking.

CONCLUSIONS

In this graduate work, an algorithm for quadruped robot is proposed. This gait model consists of various poses considering CoM points and provides a stable gait. A gait model with constant walking speed and distance is created. The results show that the displacement error increases as the walking speed increases.

Mimicking living things has been a long-standing concern in robotics research. While there are doubts about simply imitating animals, the idea can be useful if properly explained. This paper presents a biological approach to the control of quadruped robots. Observations of gravity load receptors and stimulus-response mechanisms provide some important clues for the control of quadruped robots.

In this graduate work, propose a gravity load controller incorporating a rhythmic pattern generator to control a quadruped robot. This controller is a simple yet effective control method that successfully integrates sensory feedback and oscillators to produce a rhythmic output signal.

The method used to address the problem of overcoming obstacles is to break down complex movement tasks into the division of complex motion tasks into groups of simple motions performed in known sequences that guarantee the stability of the robot. The method is to divide the robot into known sequences that guarantee the stability of the robot. This division is as follows.

The main features of this approach are that the dimension of the solution space can be reduced and complex tasks such as climbing and descending stairs can be implemented. First of all, the boundary of the solution space is the kinematic boundary of the robot's legs, which are very closed in Guara. In Guara they are very closed. Robots with more powerful actuators and wider kinematic limits can support the implementation of a library of obstacles related to the working environment of low-speed robots. The implementation of a library of obstacles relevant to the operating environment of a low-speed robot can be supported. It can be seen that a

set of simple actions can be combined according to the shape and size of the obstacles obtained from the contact keys and the kinematic limits of the robot.

To overcome the obstacle, a set of simple behaviors are combined to give the legged robot navigation skills and autonomy. In an environment with known obstacles, it gives the legged robot navigation skills and autonomy.

Obstacle. It is not clear when a more sophisticated model to accomplish the same task will be realized and implemented. The time it takes to implement and implement a more sophisticated model to accomplish the same task is unknown, but this approach would likely slow down the implementation. It does, however, allow the implementation of complex sequences of movements while maintaining the robot's balance.

REFERENCES

1. A.S. Millman three-dimensional object recognition // ACM Computing Surveys, (CSUR). - ACM Press. -2020. vol. 17. - Vol. 1. - P. 75-145.
2. Brooks R. multisensory synthesis: basics and applications with software, Iyengar S. / Prentice-Hall PTR, 2021 - 416 p.
3. V. R. Brockett, Asymptotic stability and feedback stability. Differential-geometric control theory / V.R. Brockett, H.D. Sussman, ed. - USA: Birkhduser Boston, 2020, Inc - 181-191.
4. Ching-Chang Wong, Hou-Yi Wang, Kuan-Hua Chen, Chia-Jun Yu, and H. Aoyama. Motion controller design for two-wheeled robots based on collective learning frameworks // SICE Annual Conference, Tokyo. - 2020. - P. 772-776
5. Chen, L., F. Bai, and L. Wu. Motion control of a wheeled robot based on angular velocity sensors. //IEEE Conference on Robotics, Automation and Mechatronics, September 21-24, Chengdu, China - 2021. p. 598-602.
6. Dixon, W. E., D. M. Dawson, F. Zhang, and Z. Erkan. Global exponential tracking control with PE conditions for mobile robotic systems // IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics. vol. 31(1). - 2020. p. 129-142.
7. E. Papadopoulos, I. Poulakakis, and I. Papadimitriou, "On path planning and obstacle avoidance for non-holonomic platforms with manipulators: a polynomial approach // International Journal of Robotics Research, IJRR, vol. 21, 4th - April 2020. p. 367-383.
8. Elfes A. Sonar-based real-world mapping and navigation // IEEE Trans. - 2020. - Vol. 3(3). - P. 249-265.
9. Elfes A. Using occupancy grids for mobile robot perception and navigation // Intern Journal Computer. - 2020. - Vol. 22. (6). - P. 46-57.

10. Fierro, R., and F. L. Lewis Control of non-holonomic mobile robots: A step back from kinematics to dynamics // Proceedings of the IEEE Decision and Control Conference. New Orleans, Los Angeles, USA. 2021. p. 3805-3810.
11. Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.. Stable tracking control methods for autonomous mobile robots // Proceedings of the IEEE International Conference on Robotics and Automation. Cincinnati, USA. - 2020.- P. 384-389
12. Luca, A., Oriolo, G., Samson, C., Laumont, J.P. Robot Motion Planning and Control, a chapter in the book Feedback Control of Robots with Non-Holomic Characters. Synopsis of lectures on management and informatics. Springer. 2020 - S. 328
13. S. Hashim, Tien-Fu Lu. New strategies in dynamic motion planning of non-holonomic mobile robots depending on time // IEEE International Conference on Robotics and Biomimetics (ROBIO). 2021 p. 200-250
14. Vanmarke E. Random fields: analysis and synthesis / Vanmarke E. - MIT Press, Cambridge, Massachusetts. 2021
15. Walsh, G., D. Tilbury, S. Sastry, R. Murray and J.P. Laumond. Trajectory stabilization of systems with non-holonomic constraints // IEEE Transactions on Automatic Control. 2021
16. Ichihara, Y. and K. Onishi. Trajectory planning and tracking control of a wheeled mobile robot taking into account the capabilities of the robot // IEEE International Conference on ICIT 2020. p. 104-140
17. Y. Yamamoto. Cooperative movement and manipulation of mobile manipulators / Y. Yamamoto, X. Yun. - Singapore: World Scientific Publishing Co. Ltd, 2020. p. 157-181.
18. Batanov A.F., Robotic complex for special operations // Special technique. No. 6. - 2020. - C. 10-17.

19. Murkin S. In Information about the organizational and legal sphere // Legislation of Ukraine.
20. Gritsynin S.N. Control systems for the movement of wheeled robots / Burdakov S.F., Miroshnyk I.V. - St. Petersburg: Nauka, 2021 - 229 p.
21. Stelmakov R.E. sanitary classification of work according to indicators of harmfulness and dangerous factors of the production environment, difficulty and tension of the labor process. Approved by order of the Ministry of Education and Science of Ukraine No. 528 of December 27, 2021.
22. Golovko V. Neurointelligence: theory and application. Book 2 / Golovko V. - BPI, Brest, 2021.
23. Guest 12.2.032-78 "Ssbt. Workplaces for sedentary work. General ergonomic requirements" 2020.
24. V.G. Hrade Controlled movement of mobile robots on surfaces arbitrarily oriented in space / Hradetsky V.G., Veshnikov V.B., Kalin S.V. Germany: Nauka, 2021.
25. Gurv V.A. Mathematics: reference book: 2nd edition / V.A. Gurv, A.H. Mord. - Germany: Enlightenment. 2020
26. Shver E.I. v.2.5-28-2021. engineering equipment of buildings and structures. Natural and artificial lighting.
27. E.I. Dynamics of robot management / under the editorship. - Germany: Nauka, 2021.
28. DSN 3.3.3.6.037-99. Hygienic norms of industrial noise, ultrasound and infrasound.
29. Mor F.R., Poduraev Yu.V., Mobile robots for inspection and repair of underground pipelines: current state and prospects for development // Mechatronics. 2020

30. Conducting a comparative analysis of the legislation of Ukraine and NATO member countries in the field of information with limited access / : <http://www.justinian.com.ua/article.php?id=2461>.

31. I. Pulyuya. improvement of methods of creating local maps of the mobile robot environment // Bulletin of the Ternopil State Technical University named after

32. obstacle detection method for mobile robots using heterogeneous sensor fusion technology // Automatic control systems and automation devices.

33. Topchiev Yu robotics: history and prospects / Topchiev Yu. - Germany: "Nauka" Publishing House, MAI.

34. Mak, I.M. Robotics: History and prospects / I.M. Mak, Yu.I. Topchiev, I.M. Maka, Yu.I. Topchiev - M.: MAI: Nauka Publishing House, MAI. 2020. - 101 p.

35. Shiyonov A. I Work management /. Shiyonov A. I - Germany, 2020. - 187 p.

36. Miro I.V. Nonlinear and adaptive control of complex dynamic systems. / - K.: SNB: Germany Nauka, 2020. - 325 p.

37. Okhot D.E., Plakh A.G., Tug A.N. Basic algorithms of game modeling and control of robot football players // Artificial intelligence. - No. 3. - 2020. p. 534-540.

38. Pavko A.V. New technologies for controlling the movement of technical objects // Collection. Art. according to the materials of the 7th International Congress of Scientific and Technical Robots. conf. - K.: NTUU "KPI", 2020. - 184p.

39. Yu.V. Poduraev, Mechatronics: basics, methods, applications: Textbook for students studying the "Mechatronics" specialty of the "Mechanical Engineering" preparatory course, 2020.-255p. "Mechatronics and robotics".

40. G.V. Pisme, basics of robotics. Introduction to the specialty: a textbook for universities. - M.: Higher school. - 2020. - 224 p.

41. Po. E.P. Basics of robotics: an introduction to the specialty / E.P. Po., - M.: Higher School, 2020. 224 p.
42. Ueno H.J. Representation and use of knowledge. In: Ueno H. (ed.), Representation and use of knowledge. - Trans. from English - M.: Izuzuka,; 2020. - 220 p.
43. Machu I.I. Robotic systems and complexes / I.I. Machu, edited by Prof. M.: Transport, 2020. - 464 p.
44. Spinu G.A., Industrial works, constructions and applications. Tutorial. 2nd edition / Spinu G.A. - K.: Higher school - K.: Higher school, 2020. - 311p.
45. Spinu G.A., Robots with artificial intelligence / Spinu G.A. - K.: Vyshcha Shk.: Technika, 2020. 111p.
46. Yu.D. Andrianov, Management systems of industrial robots / Yu.D. Andrianov, L.Ya. Glazer, M.B. Ignatiev and others. - M.: Mashinobuduvaniya, 2020 - 288 p.
47. Gonzalez R. Robotics / Fu K., Lee K.... - M.: Mir, 2020 - 624p.
48. Shahi M. Course of robotics / Shahinpour M. Trans. - 527 p.
49. Yure E. I. Control of robots and robotic complexes / Yure E. I. - LSU: Mechanical Engineering, M. - L.: Mechanical Engineering, M. - L.: Mechanical Engineering, 2020: Mechanical Engineering,. and. - L.: Mechanical engineering.. - L.: Mechanical engineering: Mechanical engineering, Leningrad branch, 2022. - 235 p.
50. Yaru N.G. Fundamentals of the theory of fuzzy and hybrid systems: Study guide / Yaru N.G. - K.: Finances and statistics, 2020.

APPLICATIONS

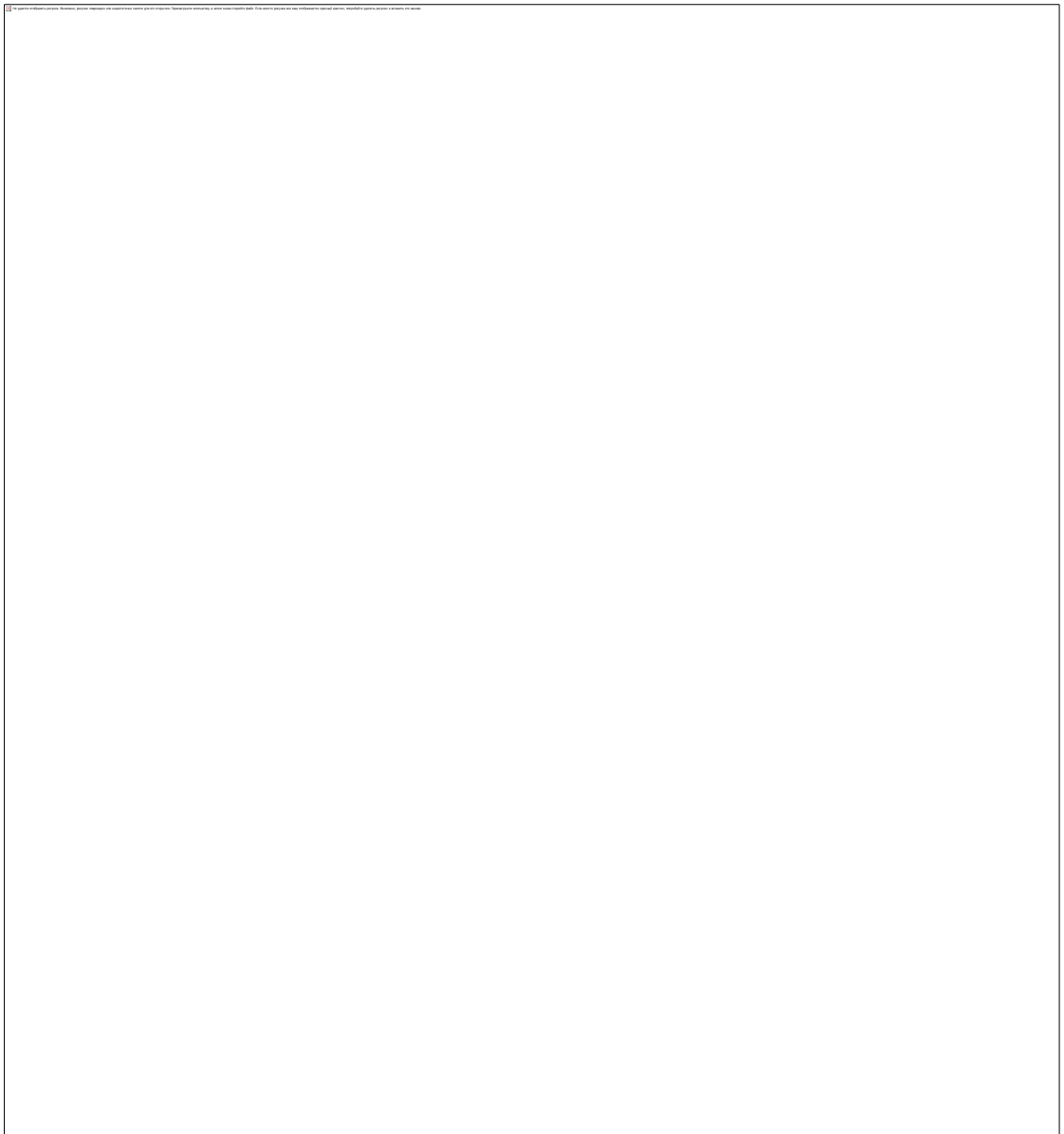
Appendix 1.1

Listing of the program code of the "main_stabilization" procedure

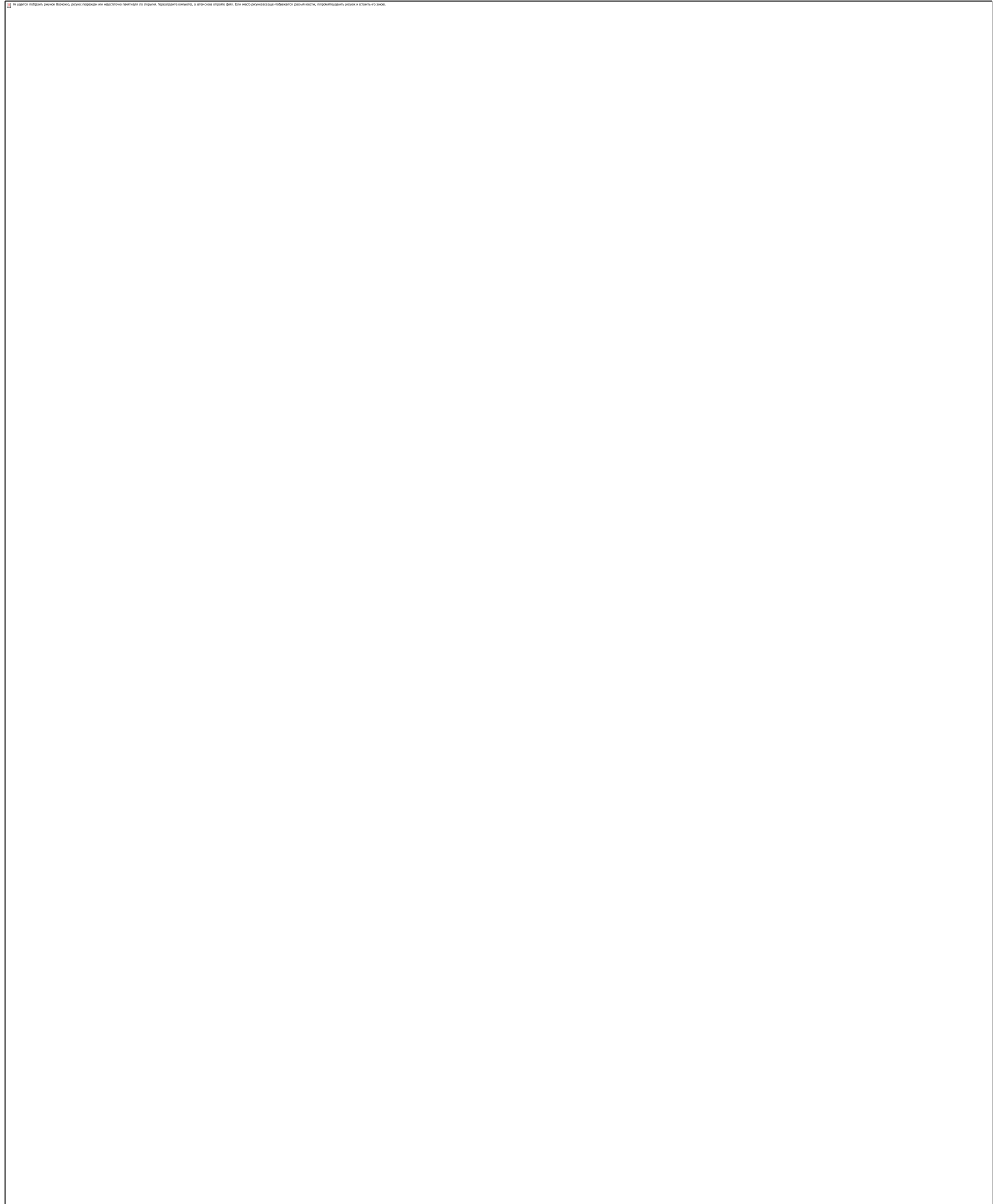
Appendix 1.2

Listing of the program code of the function fi_bilshe90

```
function fi_bilshe90 (r,l,d)
    chyselnik=(d(1)^2+d(2)^2)*sqrt((d(1)-l(1))^2+(d(2)-l(2))^2);
    znamen=(d(1)-l(1))^2+(d(2)-l(2))^2+d(1)^2+d(2)^2-l(1)^2-l(2)^2;
    DC=chyselnik/znamen
    %pause
    x_centr=d(1)+(DC*(d(1)-r(1))/sqrt((r(1)-d(1))^2+(r(2)-d(2))^2))
    y_centr=d(2)+(DC*(d(2)-r(2))/sqrt((r(1)-d(1))^2+(r(2)-d(2))^2)) plot([0 x_centr
    d(1)], [0 y_centr d(2)], 'b', 'LineWidth', 2);
```



Results of simulation modeling of stabilization of a mobile robot at $\varphi > 90^\circ$



Appendix 1.4

Listing of the program code of the function `fi_menshe90()`

