3D printed Robot Dog Walking on Terrain for STEM education

Interim Report

Lau Tsz Yin (3035686516)
22nd January 2023

Supervisor:
Dr. Chim, T.W.
Mr. David Lee
Abstract

Nowadays, quadruple robots have extensive applications in various aspects, such as element extraction, site inspection and product delivery, etc. Therefore, there is an ongoing research need on shortening the task completion time of the above tasks. This project aims to build a 3D-printed robot dog and improves its walking ability on even terrain. Walking gait and self-balancing algorithms were implemented by Arduino programming. The coding logic is based on inverse kinematics, while the inputs were collected by sensors on the robot’s joint.

At the current stage, the components of the robot dog were printed, the whole robot dog has assembled and the studies on 3D modeling and Arduino programming syntax have completed. Although some printing trials were failed at the beginning, all components were printed eventually, and the quality is more than satisfactory. The major next steps will be on implementing walking gait and self-balancing algorithms. In the future beyond this project, more research could focus on improving the walking speed, or even modeling robot ‘arms’ to identify and pick up items, and then determine the shortest or safest path to deliver them to specific areas.
Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr. Chim for the continuous support of my Final Year Project, and his patience, motivation, enthusiasm, and immense knowledge.

Secondly, I would like to express my special thanks of gratitude to Mr. Lee who gave me a lot of useful tutorials to tackle technical issues I met throughout the project. It was a great honor to work and study under his guidance.

Besides my advisors, I would like to thank my second examiner Dr. Dirk for giving me many useful advice and comments.

Last but not least, I am extremely grateful to my family and friends for their love, care, and accompany throughout my whole life.
# Table of Contents

ABSTRACT ................................................................................................................................. II

ACKNOWLEDGMENT .................................................................................................................. III

LIST OF FIGURES .................................................................................................................. V

LIST OF ABBREVIATIONS ....................................................................................................... V

1. INTRODUCTION .................................................................................................................... 1
   1.1. PROJECT BACKGROUND ............................................................................................... 1
   1.2. MOTIVATION TO THE STUDY .................................................................................... 2
   1.3. PROJECT OBJECTIVES ................................................................................................. 2
   1.4. OUTLINE OF THE REPORT ........................................................................................... 3

2. METHODOLOGY .................................................................................................................... 3
   2.1. HARDWARE TOOLS ....................................................................................................... 3
       2.1.1. Microcontroller unit ................................................................................................. 3
       2.1.2. Inertia Measurement Unit ....................................................................................... 4
   2.2. SOFTWARE DEVELOPMENT .......................................................................................... 5
       2.2.1. Arduino programming ............................................................................................. 5
       2.2.2. Android app ........................................................................................................... 5
   2.3. INVERSE KINEMATICS .................................................................................................. 5

3. RESULTS AND DISCUSSION .............................................................................................. 6
   3.1. EVALUATION OF CURRENT PROGRESS .................................................................... 6
   3.2. MAJOR NEXT STEPS .................................................................................................... 7
   3.3. ANTICIPATED DIFFICULTIES ..................................................................................... 7

4. CONCLUSION ....................................................................................................................... 8
   4.1. LIMITATIONS ............................................................................................................... 8
   4.2. RECOMMENDATIONS .................................................................................................. 8

REFERENCE ............................................................................................................................. 9
List of Figures

Figure 1: Spot P. 1
Figure 2: Microcontroller Unit P. 4
Figure 3: Inertia Measurement Unit P. 4
Figure 4: The assembled robot dog. P.6

List of Abbreviations

MCU Microcontroller Unit
IMU Inertia Measurement Unit
STEM Science, technology, engineering, and mathematics
3D Three Dimensional
CPU Central Processing Unit
IDE Integrated Development Environment
PLA Polylactic acid
AI Artificial Intelligence
1. Introduction

1.1. Project Background
Quadruple robots are defined as four-legged robots. Composed of a main body and control unit, it can not only walk as a quadruped animal but also make decisions like a human.

In the past, workers could have lost their fingers if they had careless mistakes in welding car components. Therefore, American George Dewall had created the world’s first programmable robot which can replace human in welding. [1] This marked the time where human started to think how robot can replace themselves in other aspects.

A lot of useful quadruple robots are developed from then on. For example, ANYmal C was developed by ANYbotics [2], and HyQ was produced by the Italian Institute of Technology [3]. These robots have proven that quadruple robots are an alternative or even better solution when traveling through terrains other than wheeled robots. [4]

This project uses Spot, a robot dog released by Boston Dynamics in 2019. Its transformative mobility automates routine inspection tasks more accurately [5]. For example, its junction between legs and bodies has 360° perceptions which stimulate quadruped animals’ ankles and can provide flexibility when walking on terrains. Other features will be discussed in later chapters.

Fig 1: Spot, a quadruple robot dog released by Boston Dynamics, serves as this project’s robot model
The scope of this project will focus on walking gait algorithms instead of Toit algorithms. It is because the Toit algorithm requires higher robot leg capability which is essential in the rapid self-balancing of robots. More calculations and re-modeling work will be involved. Also, this project will focus on the walking ability of robot dogs, but not aims to modify it to perform tasks, such as carrying products. Due to time constraints, around five lessons of STEM courseware will be designed in the later stage of this project.

1.2. Motivation to the study
At present, quadruple robots participate in various applications, such as inspection work, e.g., inspecting buildings for gas leakage; and delivery, e.g., transporting goods through paved roads [6]. Therefore, walking on terrains is the fundamental element for almost every application. However, it is difficult to implement an algorithm which is suitable for all roads and robots. Most algorithms programmed to the robots are set to be adaptive to specific applications, or even a specific road. Unpredicted obstacles or miscalculations may cause robots to fall into errors. Therefore, there is an ongoing need to develop more comprehensive algorithms for robots to walk and climb efficiently.

Another importance of this project is related to STEM education development. As every child is a possible future engineer, this project includes mathematics (inverse kinematics calculations), hands-on experience (robot building with hardware tools), and computer programming (Arduino programming). This project will be used as a skeleton for creating related STEM teaching materials. Developing children’s interest in this topic will continue the optimization of robot dogs in the future.

1.3. Project Objectives
This project aims to improve the performance of 3D-printed robot dogs walking on uneven terrain. By implementing robot dog inverse kinematics, together with walking gait and self-balance algorithm, the robot dogs will be capable of walking across rocks and can climb up and down.

A mobile app will be developed to control the robot dog. And a set of STEM course materials will be constructed for this project.
In other words, the project deliverable includes:
- A fully functional robot dog that can walk on terrain
- Code on robot dog by Arduino programming
- An Android app as a controller for a robot dog
- A set of courseware and teaching materials using this robot in STEM education

1.4. Outline of the report
The remainder of this paper proceeds as follows. Chapter 2 discusses the methodology of this project, including hardware tools such as Microcontroller unit which acts as a central control system in robot, and software development platforms like Arduino programming in robotic system. Chapter 3 discusses the current project status and the major next steps. Chapter 4 concludes this report and proposes recommendations.

2. Methodology
This chapter is separated into three sections: hardware design and tools, software programming languages and physics logic.

2.1. Hardware Tools
2.1.1. Microcontroller unit
Microcontroller unit (MCU) can be viewed as a small computer on a single integrated circuit [7]. It contains one or more CPUs, program memory storage, and an input/output interface. It is widely used in embedded systems, for example, automobile control systems, and implantable medical devices.

In this project, MCU is used as the main control system of the robot dog. According to Figure 2, after receiving the input order, it will control the movement of servomotors that are connected to the robot dog legs. Different algorithms can also be stored in it and run with the help of MCU. The buzzer is to ensure programs are runnable in debugging and the regulator can maintain a stable voltage input to servomotors. Since MCU acts as the “brain” of the robot dog, it is the container of Arduino programs and the robot dog’s actions.
2.1.2. **Inertia Measurement Unit**

Inertia Measurement Unit (IMU) is a 9-axis sensor that measures velocity, orientation, and gravitational forces [8]. It detects rotational movement of the three-axis: Pitch, Roll, and Yaw, while the Accelerometer, Gyroscope and Magnetometer are inserted into it. The common models of IMU are MPU-9250 and MPU-6650.

In this project, the model used is MPU-9250 (see Figure 3). It can be used to detect the rotational movement of four robot dogs, and the data obtained will be used in inverse kinematics calculations.
In other words, MCU and IMU cooperate closely: MCU controls the start of IMU, and the position data collected by IMU will pass to MCU for processing and deciding the further movement of the robot joint.

2.2. Software Development

2.2.1. Arduino programming
Arduino code is based on C++, and on top of that special methods and functions are added [9]. Arduino Integrated Development Environment (IDE) is used to write Arduino programs. It compiles the code and translates them to Arduino executable code.

2.2.2. Android app
An android app will be created for testing and debugging. It controls the start, and the overall movement of the robot, etc. It will connect to MCU by Bluetooth. The IDE to develop this Android app is Android Studio and the programming language is Java Version 8.

2.3. Inverse kinematics
Inverse kinematics is the opposite algorithm of forward kinematics. The input of it is the target position, then the inverse kinematics algorithm will calculate the pose or action required for the end effector (merely the position at the end of the robot dog’s legs) to arrive at that target position [10]. Therefore, the output is the angles. This algorithm is applied to this project as the coordinates of robot logs are measured by servomotors, while the pose of the dog’s legs to reach target position is known.

In conclusion, hardware tools such as MCU and IMU will be embedded inside the robot dog and initialized first. Then Arduino code will be written based on Inverse kinematics which specifies the physical logic of the robot leg’s movement.
3. Results and Discussion

This chapter illustrates the current working progress as of 30th November 2022.

3.1. Evaluation of Current Progress

The progress is acceptable. More effort should be added to tackle future unexpected difficulties.

All components of the robot dog were printed in HKU InnoWing, although some 3D printing trials failed due to a lack of support or printer malfunction. These unexpected problems are unavoidable. The main body part which contains MCU is printed in black color while other parts, for example, the right joint, and left shoulders, are printed in white. The infill of 3D printed components is in cubic size with 20% infill capacity. There are “tree supports” which prevent overhanging of PLA materials while printing and they are removed afterward. Overall, the quality of the 3D printing components is more than satisfactory as there are optimization algorithms by the 3D printer in HKU InnoWing.

Also, assembling the robot dog was completed. The surface of the 3D-printed components was polished to ensure a smooth assembly of components. By following the instructions booklet and using different sizes of nails, a complete robot dog was assembled. It is verified that when the switch of the MCU is turned on, a bumping sound is heard by the buzzer (see Figure 2).

![Fig 4: The assembled robot dog (a)in side view (b)in front view (c)in vertical view](image-url)
3.2. Major Next Steps

Next, Arduino programming work will be started. The coding environment will be set up and the GitHub repository will be landed. By searching for reference materials of Arduino syntax and studying the existing code, the walking gait algorithm can be coded.

For implementing walking gait algorithm, the input is the joystick value entered in the mobile app. After value mapping, it will enter to a function which do inverse kinematics calculations. Three angles of each leg will be outputted. Next, control the servomotors and adjust them according to the angles. With coordination of four legs, a “crawl” walking mode cannot be created.

After implementing walking gait algorithm, self-balancing algorithm will be developed, with the initialization of IMU. The study on IMU has not started yet. The main direction is the have a balancing step after each leg’s movement.

3.3. Anticipated Difficulties

It is expected that the coding of the walking gait algorithm is hard. It is because there are limited resources about Arduino programming on the internet. For debugging process, extra testing functions should be developed to find out where the problem exists. In other words, debugging and refactoring processes will be time-consuming. Therefore, this project may review more online source code so that the coding logic implemented can be clearer.
4. Conclusion

Quadruple robot learning is an AI subject undergoing intense study. It can not only replace a lot of repeated work that humans can do but also complete dangerous works which humans cannot do. In this project, a 3D-printed robot is being built that can walk on even terrain, by implementing walking gait algorithms and self-balancing algorithms.

The main action to complete the technical part of this project is: first complete 3D printing and assemble the robot dogs, next implement the two algorithms and finally testing and debugging. At the current stage, 3D printing is completed and assembled. The quality of printed material is satisfactory, and less polishing is needed afterward. It is very important to have an accurate and well-polished component because they determine the limit of the greatest speed the robot dog can afford. The infill PLA material is 20% in cubic form, which strikes a balance between the stability of its balancing ability and the printing hours needed. Therefore, the printing work done in October will directly contribute to the later testing and debugging work.

4.1. Limitations

This project will only implement the walking gait algorithm, but not the Toit algorithms. It is because implementing the Toit algorithm requires review and analysis of existing hardware tools, the mechanical structure of the robot dogs, etc. Due to time constraints, it cannot be added within one-year time. Other than that, the robot dog does not contain a robot arm that can be assigned to do other tasks, such as distributing items to specific areas. It also requires the rebuilding of existing robot structures. Moreover, when the robot walks on uneven terrain, it does not visualize the road and evaluates which path will have fewer obstacles. The actual optimal solution is to find a path which has fewer barriers and a shorter path. It requires AI searching techniques implemented in the IMU, which makes the project much harder.

4.2. Recommendations

Based on the limitations written in Section 4.1 Limitations, a few recommendations are suggested. First, review the completed work and evaluate whether it can implement the Toit algorithm, as well as remodel some parts of the robot to add one or two arms. Next, review the IMU computing power and evaluate whether AI search algorithms can be added which make the robot finds the “best” path instead of walking randomly and solely rely on its self-balancing power.
Reference


