Polydog – Quadruped Robot, with a Three-degree-of-freedom Parallelogram System

Hugo Durand, Younes Bazi, Raphaël Anjou

***Abstract*—This article presents the development and implementation of a robot dog called Polydog. Several versions have been created, according to our vision and understanding of the robot. It's equipped of a motor driver SSC32 powering 12 servomotor TD8135MG controlled by a Esplora remote via Bluetooth. The article highlights the process of a dog robot's development and the limitations of electronics components and of some methods.**

***Index Terms*—Robot dog, legged robot, parallelogram system, Arduino, servomotor, PlateformIO.**

I. Introduction

THE use of robot dogs, also known as quadruped robots, offers several advantages over traditional wheeled robots in difficult terrain. It is known that's a subject of research and competition between large companies such as Boston Dynamics or Unitree Robotics which invest in this kind of project because they see in it an incredible automation of tasks in the future. They can be trained to perform a variety of tasks, such as assisting people with disabilities, perform boring or dangerous tasks for humans, and even serving as a security measure. On our scale we aim to develop low-cost, DIY and an open-source version that can be built in any fablab, without any increased computer knowledge. The advantage of their lower cost is that they are more easily inclined to be sent for small valuable missions, such as reconnaissance in the army without fear of a big loss.

The rest of this article is organized as follows. In Section II, the designed research of a leg and the construction of the robot. In Section III, the electronic components used inside the Polydog. In Section IV, we present the algorithm to bring it to life. In the final Section, the conclusions are presented.

II. Building the robot

A. Parallelogram system

Based on Nina Schaller's research on the walking of an ostrich and the bones composition [1], we opted for a modeling of the leg integrating the parallelogram system which allows to keep the mass of the leg closer to the body and limit the inertia of the leg.

H. A. N. Durand, University Côte d'Azur, Robotics Department, 930 Route des Colles, 06410 Biot (e-mail: hugo.durand@etu.univ-cotedazur.fr).

Y. Bazi, University Côte d'Azur, Robotics Department, 930 Route des Colles, 06410 Biot (e-mail: younes.bazi@etu.univ-cotedazur.fr).

R. Anjou, University Côte d'Azur, Computer Science Department, 930 Route des Colles, 06410 Biot (e-mail : raphael.anjou@etu.univ-cotedazur.fr).

We had now to build a body light enough to support. We tested two solutions, in 3D printing first and then in aluminum. It turns out that the weight did not decrease but the body is much stronger in aluminum.

Fig. 1. Legs with 3 degrees of freedom. A servo fixed to the body rotates the cylinder attached to the other end by a ball bearing; the second one moves the whole thing back and forth through the gear and the last one controls the lower part with the parallelogram system.

We have tested several types of gears. In the first robot version [2] , we tried spur & helical gears, modifying the module (reduction of the pitch respecting the precision limits of a 3D printer nozzle) and the gear ratios (increasing the output torque while keeping a sufficient amplitude of movement). In the latest version, we opted for herringbone gear [3] with a 1.3 mm module and 30 teeth sprockets. Compared with single-helical gears, the herringbone gears have significant advantages of higher load carrying capacity, lager total contact ratio and lower axial force.

B. Modeling and printing

We modeled the whole animal on the Onshape software (used in Peip1 course), facilitating the management of the links between the 8 designed parts per leg (3D printing, PETG, 20% filling), the body (made in aluminum) and the servo horns. The feet are made of ninja flex filament, with a rubber coating, but they can still be improved to grip the ground more.

In order to reduce the play of the legs, we have developed our own servo horn, by reproducing the pinion of the head, with resin. It has the form of a propeller with 3 blades allowing to embed it in the support box of the servo motors.

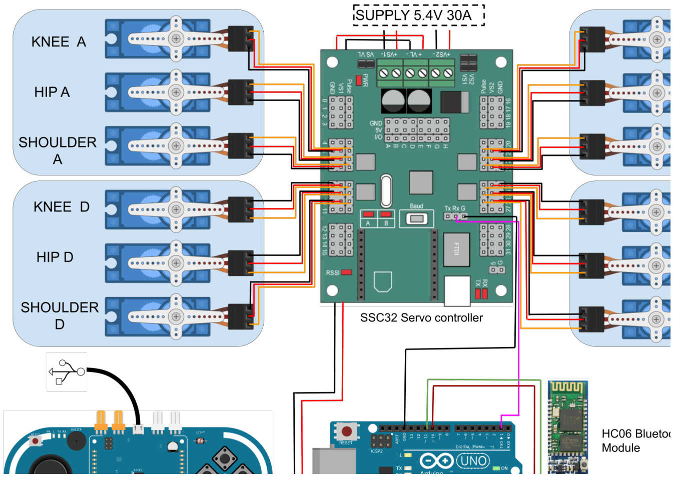

Fig. 2. Electronic diagram of the robot. This allows to give a global vision of the whole and to observe the links between the different components. This has been designed on Draw.IO [4].

III. ELECTRONIC COMPONENTS

A. *The central control unit*

The Polydog uses an Arduino Uno microcontroller as its central processing unit, which allows for the integration of various sensors and actuators. It is however limited in the number of 5v and GND pins, currently saturated on our robot. Moreover, as we only have two Rx Tx I/O, we used the SoftwareSerial library replicating the same functionality.

B. *Motors and driver*

The robot dog is equipped with 12 TD8135MG servo motors 35kg/cm of torque, which are controlled by a motor driver SSC 32. The reading of the user's guide was essential for us to understand the functioning of the card, especially the power options (vS1, VS2, VL). We supply the logic circuit VL with the Arduino board and we connect the two power channels VS1 and VS2 under 5.4 V for the servo motors with a power supply providing up to 50A [5].

After noticing momentary voltage drops, to which the addition of capacitors was not enough to store enough electrical energy to overcome the electrical resistance of the cable. By increasing the cross-section of the power cable and manually changing the output voltage of the power supply, we solved the servo operation problems.

C. *Esplora board controller*

Finally, we used an Esplora remote to control the Polydog, which was equipped with the Bluetooth module Hc06 (Slave). The module Bluetooth Hc05 (Master) [6] connected to the Arduino, via software pins 10 & 11.

The robotic dog was programmed to respond to various inputs from the Esplora controller, such as button presses, joystick movements, and sensor readings. For example, we can tilt the body of the robot back and forth, to one side or the other, following the values of the accelerometer on the remote.

IV. ALGORITHMS

The algorithm starts with the commands sent to the ssc32 servo controller, via the serial communication by sending string data of this type: "#<motor_pin>P<angle between 500-2500 microseconds>T<time in milliseconds>" from the Arduino Uno and converts them into control signals for the servomotors.

We have implemented a basic function that manages the sending by adapting the offsets depending on the orientation of the servo motor in a class named CustomServo (servoAH=0° versus servoBH=180° for the same position, for example). We, then build a Leg class to instantiate each leg and correct imprecisions between the legs and make sure that the 4 legs form a rectangle on a table.

PolyDog class record empirically constructed movement patterns such as trot_walk(), stand_up(), composed by a list of position command in this form: legA.move_hip(95);

We chose to code with PlatformIO, an open-source IDE for embedded systems that aims to make it easier to create, debug, and implement projects like this. It offers features such as good dependency management to simplify the installation of necessary libraries like Esplora and an integrated debugger.

V. CONCLUSION

Overall, the development of a robot dog is a challenging task starting in the art of maintaining it standing on 4 legs, then 3 legs and finally 2 so that it can move by biomimetic imitation like four-legged animals. While the current Polydog is capable of walking, it limps a little, which means there is still room for improvement. With further research and development, it is expected that the Polydog will become more stable, efficient and capable of performing more complex tasks.

REFERENCES

[1] N. Schaller, “Birds on the run: what makes ostriches so fast?” in the scienceinschool.org website, November 22, 2011

[2] R. Anjou, H. Durand, Polydog_version1 in Instructables, a website specialized in DIY projects, https://www.instructables.com/Polydog-Arduino-Project/

[3] R Keith Mobley, 39 - Gears and Gearboxes, in Plant Engineer's Handbook, 2001

[4] Gaudenz Alder, Draw. Io, diagrams.net, a free and open-source cross-platform graphic drawing software developed in HTML5 and JavaScript.

[5] Lynxmotion_ssc-32u_usb_user_guide

[6] Pascal Masson, “Eléments de robotique avec Arduino, Communications RF”, http://users.polytech.unice.fr/~pmasson/Enseignement.htm

Durand Hugo received high school diploma from the French high school in Cali, Colombia. Since 2020, he’s student at Polytech Nice Sophia, currently in the Robotics Department.