gribot

Specification for an open source agricultural robotics platform

gribot: Specification for an open source agricultural robotics platform

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1. Introduction

In the requirement document, we described *what* the gribot platform should do and the goal to reach. The specification document describes how we will do it. This document is much more technical as it contains many subjects such as:

- Analysis
- Calculations and dimensioning
- Technology choices
- etc.

During the preparation of the specification document, it is often necessary to revisit the requirements definition document to clarify a point, and sometimes even to modify an unclear specification, or to add one to clarify the situation.

In this document, specification will be split into two main categories:

Platform specification	These specification concerns the whole *gribot* platform, such as general architecture, software to use, technology to use, etc. In other word, all elements common to the *gribot* robot family. As an example, the navigation system concerns the whole platform.
Robot specification	These specification concerns only one robot in the family, performing a specific function. It contains only those elements that are useful for this robot. As an example, the mower cutterbar does not concern the weeding robot.

The following chapters will describe these different specifications.

Chapter 2. Platform specification

The platform specification chapter contains element common to all robot that can be derived from the *gribot* platform. It goes from the propulsion system to navigation, including communication between the various elements.

Mechanical design

The mechanical design is specific to the robot that will be built.

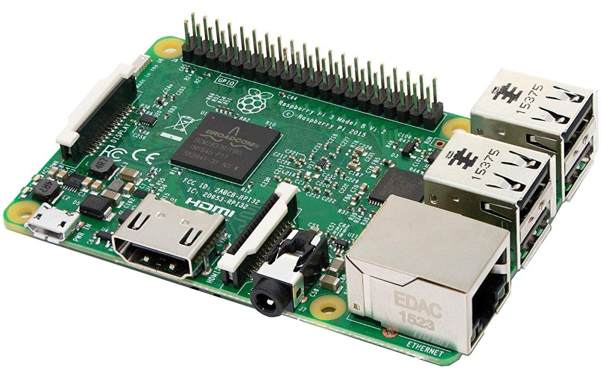
Embedded computer

The embedded computer is one of the critical elements of the robot. He is in charge of hosting the software that manages the robot. The embedded computer must satisfy several constraints:

- It must be affordable
- It must have enough memory to allow the operation of the various embedded software packages.
- Its disk capacity must be sufficient for the storage of the files necessary for the robot's operation.
- It should have a WiFi interface.
- It must have a family of interfaces, including CAN, I2C, USB, serial, etc. Either directly on the card or via plug-in cards.
- I must have interfaces or navigation charts such as GPS, accelerometers, electronic compasses, etc.
- It must support Ubuntu distribution.

Raspberry Pi is one of these small computer that fulfill almost all of the requirements above. At the time of writing, the available version is the Raspberry Pi 3.

Figure 2.1. Raspberry Pi



The Raspberry Pi 3 has enougth memory to run Ubuntu distribution together with ROS. It also supports high-capacity SD cards, and has a whole collection of interfaces and various cards.

Communication bus

Les différents composants électronique doivent pouvoir communiquer de manière fiable avec l'ordinateur embarqué. Il existe plusieurs solutions de communication:

Star communication
: Each component is connected individually to the central computer. This solution requires one connection cable per element as well as a dedicated interface on the central computer. The number of peripheral elements and the extension of the system is limited by the interfaces available on the computer.

Bus communication
: Data bus communication may be a little more complex to implement than star communication, but it offers much greater flexibility. It requires only a small number of interfaces, and the number of systems connected to the bus, and therefore its extension possibilities, are limited only by the technology used. The current buses make it easy to connect several tenths of peripheral equipment, which is more than enough for our needs.

We will retain the communication bus for our application. However, it is possible to connect some equipment directly to the central computer if necessary, provided that the interfaces are available.

The communication bus must comply with the following criteria:

- The bus must allow the prioritization of messages.
- The bus must allow real-time communication.
- The bus must be a standard on the market, in order to be able to connect various types of equipment.
- The bus must be reliable.
- The bus must be cheap.

Among the various buses available on the market, the CAN (Controller Area Network) bus is one that meets the above criteria. It comes from the automotive world and is widely distributed. The use of the CANOpen protocol allows the connection of a large number of equipment that support this standard.

Figure 2.2. CanOPEN logo

Development of the CAN bus started in 1983 at Robert Bosch GmbH. The protocol was officially released in 1986 at the Society of Automotive Engineers (SAE) conference in Detroit, Michigan. The CAN bus is not only limited to the automobile. Due to its characteristics, it is widely used in many sectors such as industrial automation, aviation, etc.

Propulsion engine controller

To be done

Accessories engine controller

To be done

Lidar

To be done

GPS

To be done

Local contacts

To be done

Battery

To be done

Attitude sensors

To be done

Communication

To be done (WiFi, GSM, IP, etc.)

Software

Robot software

After some research, we decided to use Robot Operating System (ROS) as the management software for the gribot platform. ROS is an open source platform with a large community. This software is used by many robots, both in the research field and for production robots. On the other hand, the ROS platform has a very large number of modules and interfaces, which reduces development times.

Thanks to its modular design and communication bus, ROS makes it easy to develop missing modules: part of the software is installed on the robot, and the other part under development is installed on the developer's workstation. It is even theoretically possible that the developer may be located at a distance, in another city, another country or even another continent, from the development robot[1].

Operating system

At the time of writing, ROS is only supported on the Ubuntu operating system. However, this is not a major limitation given the very large installed base of this OS. There are Ubuntu distributions for a wide variety of platforms from servers to workstations, from embedded systems such as Raspberry Pi, Beaglebone, etc.

[1] We have not tested the feasibility of this way of developing at the time of writing.

Chapter 3. Mower

The first version of the mower robot will be of a fairly simple design. It will have 4 fixed wheels and will turn a little like a tracked machine.

Specification

The mower specification are the following:

1. Weight: 25Kg (+- 20%).
2. Delimitation of the surface to be treated by GPS RTK (Real Time Kinematic), to avoid the installation of a wire.
3. Working speed: 1,8 km/h, that is 0,5 m/s. This seems to be a reasonable speed for a small vehicle on an irregular surface.
4. Travel speed: 1,8 km/h (see comment above).
5. Ground clearance (chassis to ground distance) min 10cm, to allow movement on an uneven surface.

Propulsion sizing

Maximum resistance to advancement

We can calculate the maximum resistance to advancement force with Equation A.6, “Total resistance”. With the following values:

C_{rr}: 0.3

m: 20kg

Slope: 45°

The maximum resistance is 180N.

Wheel diameter

The diameter of the wheels is determined by the ground clearance of the robot mower chassis. Wheels with a diameter of 30cm are choosen: this ensure the minimum ground clearance and leaves some room for the layout of the electric propulsion motors. In addition, such wheels allow the use of inflated tires (at low pressure) wich makes it easier to travel on slighly uneven ground such as in crops.

The relation between the rotational speed and wheel radius is given by:

Equation 3.1. Speed as a function of rotation speed

$V = 2 \cdot \pi \cdot r . \omega$

where:

V: speed in m/s

r: wheel radius in m

ω: rotational speed in rad/s

The rotational speed can be calculated from the linear speed as follows:

Equation 3.2. Rotational speed as a function of speed

$$\omega = \frac{V}{r}$$

or, in our case:

$$\omega = \frac{0,5}{0,15} = 3,33 rd/s = 1,05 t/s = 63 t/min$$

Propulsion torque

The torque required depends directly on the diameter of the wheels and the maximum resistance force. Power, on the other hand, depends on the torque and rotational speed of the wheels, or on the forward speed and resistance force.

The total maximum force required for the movement is 180N. However, it is evenly distributed over all 4-wheel drive. It is therefore 45N per wheel.

Equation 3.3. Maximum torque per motor

$$T = F_{rr} \cdot r = 45 \cdot 0,15 = 6,75 Nm$$

The torque of each of the 4 electric motors is therefore **6.75Nm**.

Propulsion power

The total power required to propel the mower, according to the basic specifications, is:

Equation 3.4. Required power

$$P = F_{rtot} \cdot V = 180 \cdot 0.5 = 90W$$

With 4 propulsion engines, each engine must have a power of **22.5W#25W** per engine.[1]

Table 3.1. Characteristics of propulsion engines

	Value	**Comment**
Power	25W	
Torque	6,75Nm	
Rotational speed	60t/min	with gear box
Power supply	12 ou 24 V	

Power for accessories

We reserve 180W of power for connecting accessories, such as cutter bar, arm, etc. to the robot.

[1]with the assumption that the motors and their controls have an efficiency close to 100%.

Battery pack

To estimate the battery capacity, we start from the robot's autonomy, 4 hours, and the total electrical power required, namely the power of the accessories and the propulsion power.

- Autonomy: 4h
- Total power: $90W + 180W = 270W$

12V or 24V batteries are possible. The necessary capacities are:

Table 3.2. Battery capacity

Nominal voltage	Power	Current	Capacity
12V	270W	22.5A	90Ah
24V	270W	11.25A	45Ah

Warning

In the above estimates, we did not take into account the efficiency of motors, electronic circuits, etc. However, we have calculated the necessary capacity using the maximum continuous power, which will not always be the case.

Propulsion motor

After some research, we found the following Nema 23 bldc motor 120W 24V gear brushless dc motor planetary reduction gearbox ratio 50:1 with Aliexpress.

The power is much higher than we calculated, but it was quickly available at an affordable cost.

Figure 3.1. Propulsion motor

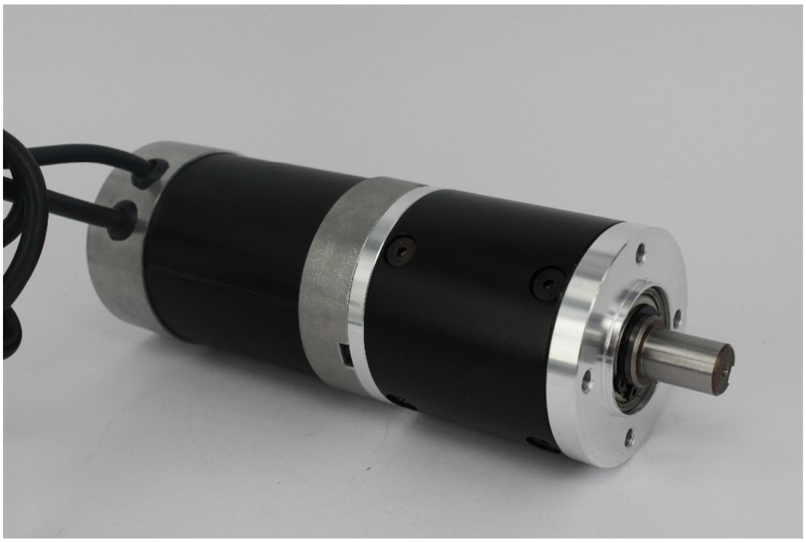
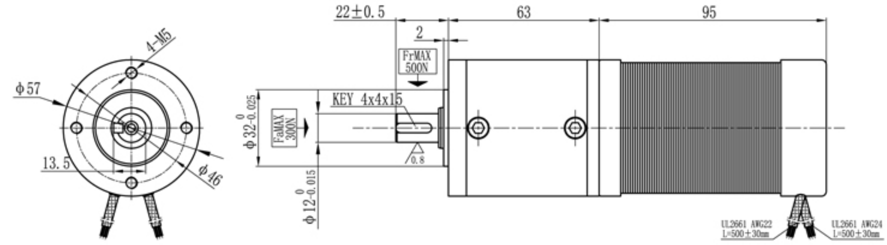
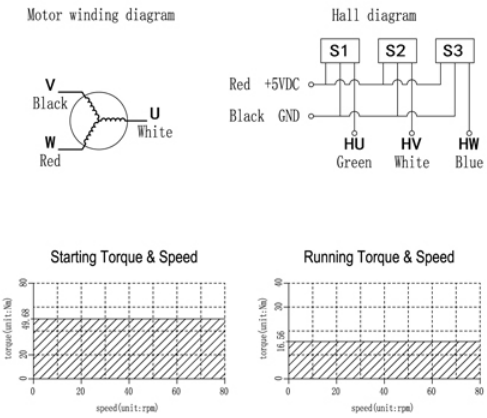


Figure 3.2. Characteristics of the propulsion motor



Technical Specificaions		
ITEM	Unit	Info
Voltage	VDC	24
Rated power	W	120
Ratio		1:50
No load speed	rpm	100
Rated speed	rpm	80
Rated current	A	8
Peak current	A	24
Resistance	Ω	0.45±10%
Inductance	mH	0.55±20%
Poles		4
Rated torque	N·m	16.56
Peak torque	N·m	49.68
Backlash	arcmin	≤25
Rotor inertia	g·cm^2	249
Voltage constant	V_{rms}/K_{rpm}	4.8
Efficiency		ca. 92%
Noise	dB	≤52
Sealing		ball bearing
Protective system		IP 40
Lubrication		life grease
Operating temperature	℃	-25~+50
Storage temperature	℃	-25~+80
Method of working		S1
Life time	h	≥5000
Weight	g	ca. 1670

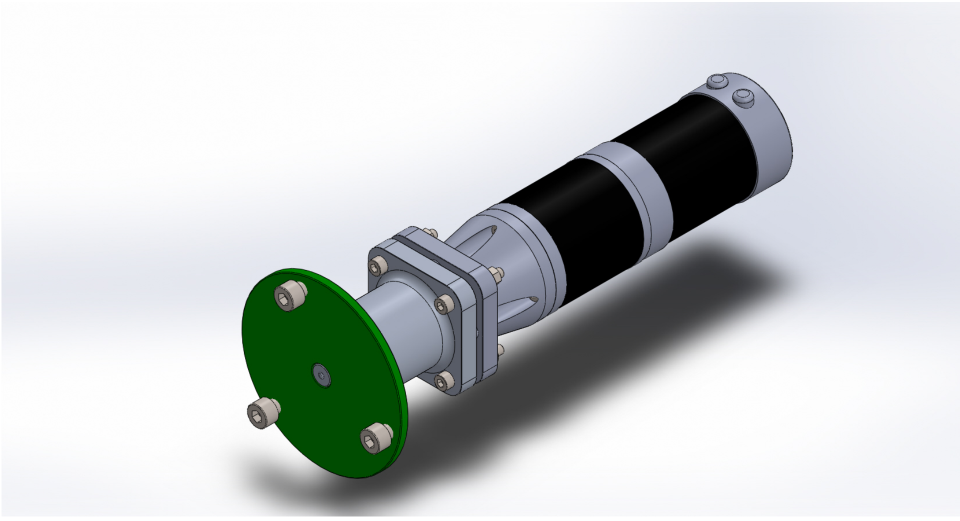


PN: PB61250391

Design

Transmission

Figure 3.3. Transmission prototype



Appendix A. Appendix

Resistance to advancement

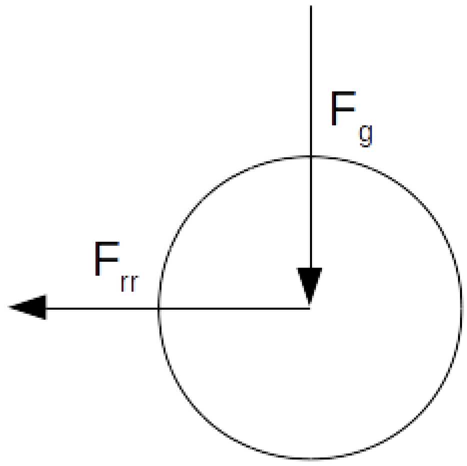
Wheel resistance to driving

The forward speed resistance is one of the parameters required to calculate the propulsion power of a wheeled vehicle. The two main parameters that influence the resistance to advancement are:

- The wheel forward resistance.
- The slope to be crossed.

According to [wiki01], rolling resistance calculations can be simplified if the vehicle does not move fast, which is our case. Not having found the rolling resistance coefficient for a tire on grass, we take the sand coefficient, 0.3, which is probably higher than that of grass. However, since the robot will move on uneven ground, which is not very similar to grass, this hypothesis leaves us a little margin.

Figure A.1. Forces on a wheel



The rolling resistance is given by:

Equation A.1. Rolling resistance

$F_{rr} = C_{rr} \cdot F_g$

Equation A.2. Gravitational force

$F_g = m \cdot g$

where

F_{rr}: rolling resistance force in N.

C_{rr}: rolling resistance coefficient.

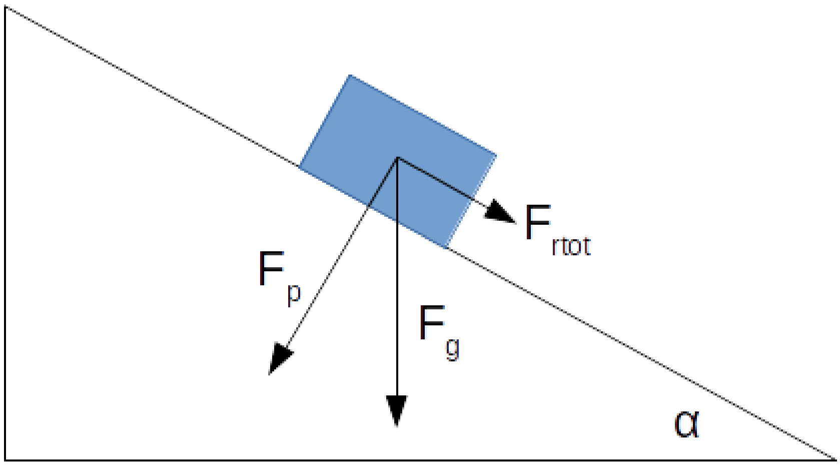
m: .mass of the vehicle (the robot for us) in Kg.

g: gravitational constant, in m/s^2.

Total resistance

Taking into account the slope, the calculation is a little more complex: it is necessary to add the components due to gravitation. In addition, the rolling resistance force decreases with increasing slope.

Figure A.2. Total resistance



The total resistance is due to the force of gravity and rolling resistance, which is:

$F_{rtot} = F_{rr} + F_{rg}$

where

F_{rr} is the rolling resistance force.

F_{rg} is the resistance force due to gravitation.

F_{rot} is the total resistance force: the one that the motors must overcome.

From the diagram and the triangle of forces, we calculate:

Equation A.3. Rolling resistance

$F_{rr} = C_{rr} \cdot F_p = C_{rr} \cdot F_g \cdot \cos(\alpha)$

Equation A.4. Gravity resistance

$F_{rg} = F_g \cdot \sin(\alpha)$

Equation A.5. Gravitational force

$F_g = m \cdot g$

by replacing the terms, you get:

Equation A.6. Total resistance

$F_{rtot} = C_{rr} \cdot F_g \cdot \cos(\alpha) + F_g \sin(\alpha) = F_g (C_{rr} \cdot \cos(\alpha) + \sin(\alpha)) = m \cdot g . (C_{rr} \cdot \cos(\alpha) + sin(\alpha))$

Chapter 4. Bibliography

Bibliography

[wiki01] *Rolling resistance*. Wikipedia.

[ROS01] *Robot Operating System*. ROS.

[Joseph2017] *ROS Robotics Projects*. Build a variety of awesome robots that can see, sense, move, and do a lot more using the powerful Robot Operating System. Packt Publishing Ltd.. Joseph Lentin. 978-1-78355-471-3. Copyright © 2017.

[Carol2016] *ROS Robotics By Example*. Bring life to your robot using ROS robotic application. Packt Publishing Ltd.. Carol Fairchild and Dr.. Thomas L. Harman. 978-1-78217-519-3. Copyright © 2016.

[ben2016] *VESC – Open Source ESC*.

Glossary

GPS RTK	Real Time Kinematic (RTK) is a satellite positioning technique based on the use of phase measurements of the carrier waves of signals emitted by GPS, GLONASS or Galileo systems. A reference station provides real-time corrections to achieve an accuracy in the centimeter range. In the particular case of GPS, the system is then called Carrier-Phase Enhancement or CPGPS.
SLAM	Simultaneous Localization And Mapping.
GPL	GNU General Public License