

OMNI-DIRECTIONAL ROBOT

Marcin Skokoń, Dominik Lutley, Marcin Wilk, Abdullah Said

*AGH Akademia Górniczo-Hutnicza im. Stanisława Staszica
Wydział Inżynierii Mechanicznej i Robotyki,
Al. Mickiewicza 30, 30-059 Kraków
skokon@student.agh.edu.pl, mwilk@student.agh.edu.pl,
dlutley@student.agh.edu.pl*

KEY WORDS: omnidirectional robot,

1. INTRODUCTION

Our goal is to create a robot that travels in all directions, avoids obstacles and follows a prepared path. Because of this, we need to account for many design requirements and make many decisions about how best to design the robot.

One important aspect for the robot is the wheels. We want to create a robot that can move in any direction, and there are two main solutions for the wheels. The first option is to use omni wheels, which have rollers perpendicular to the axis of the wheel, allowing the wheel to move sideways unobstructed. The second option are mecanum wheels, which have rollers placed diagonally (generally at an angle of 45°), so that the wheel can transfer some energy to the rollers while rotating. [1]

Based on the choice of wheel, the design of the robot needs to be adjusted to ensure proper behavior. Omni wheels should be mounted so that they are perpendicular to the robot's center. In that way they can be activated in different configurations to make the robot move in different directions. Mecanum wheels, on the other hand, can be mounted like normal wheels, but are able to produce omnidirectional motion just like omniwheels. [1,2]

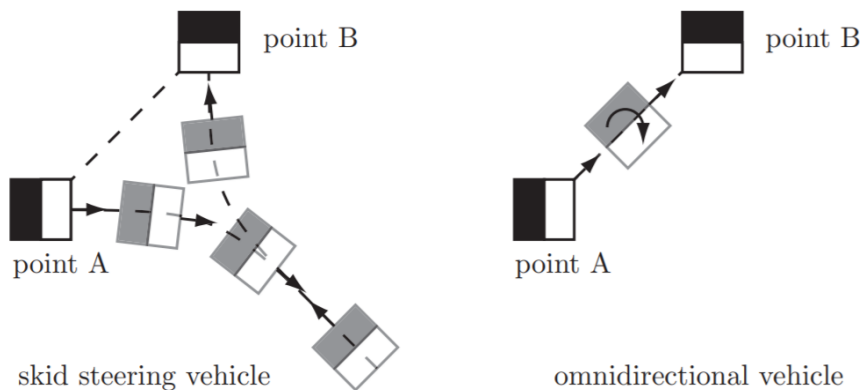


Figure 1 [3]

Below is a morphological chart, which summarizes some of the important design decisions we need to make so that the robot will function correctly. A lot of these decisions are quite important, and if we choose the wrong one it may cause the robot to underperform, or not work at all.

MORPHOLOGICAL CHART (omni-wheel robot)

SOLUTIONS						
No.	Problem	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
MECHANICS						
1	Number of wheels	3	4	5	6	7
2	Wheel configuration	Parallel	Radial			
3	Type of wheels	Omni wheels	Mecanum wheels			
4	Size of robot	Small	Medium	Large	Considerable	Massive
5	Weight of robot	Light	Medium	Heavy		
6	Case Material	Steel	Aluminum	Wood	Plastic	Lego
	Weight	Heavy	Medium	Light	Light	Light
	Strength	Very Strong	Strong	Weak	Medium	Medium
	Price	\$\$	\$\$\$	\$	\$	\$\$
7	Gearbox	None (1-gear)	2-gear	3-gear	4-gear	6-gear
8	Gearbox type	Manual	Manual with clutch	Automatic (DCT)	Automatic (CVT)	Automatic (AMT)
9	Number of gearboxes	1	2	1 for each wheel		
10	Gear ratio	High (high torque)	Medium	Low (high speed)		
11	Number of motors	1	2	1 for each wheel		
Software						
1	OS	Windows	Linux	Mac OS	Chrome OS	

2	Programing language	Python	C	C++	C#	Matlab
3	Programing Software	Dev C++	Microsoft Visual studio	Notepad ++	Arduino IDE	
Electronics						
1	Main Controller	PC	Arduino Uno	Raspberry Pi	Arduino Leonardo	More Advanced
1.1	Amount of pins	Additional microcontroll ers	Amount set by main controller	Amount set by main controller	Amount set by main controller	Depends on controller
2	Engine control	diode+transis tor	L293D controller	L298N	DR10002 DFROBOT	
3	Obstacle avoiding	Bumper	Ultrasonic sensor HC-SR04	Movement sensors		
4	Bluetooth receiver	nRF52832	HC-05	XM-15B		
5	Bumper sensor	FSR402	SEN-11999	none		
6	Path following	Pair of infrared sensors	3 or more infrared sensors	camera		
6.1	IR sensors	none	Geekcreit® IR Infrared	Iduino ST1140	DRI0002 DFROBOT	Sharp GP2Y0A41SK0 F
6.1.1	cost	0	8.22 PLN	7.70 PLN	21.03 PLN	15 PLN
7	Pins management	Bigger main controller	Arduino/Rasp berry shield	PCF8574N	MCP23017	

In red are marked chosen solutions.

1) Mechanics

The first choice is about the wheels. After some consideration, we chose to go with 4 mecanum wheels in a parallel configuration. We also considered having omni wheels in a radial configuration, which would offer the same versatility of movement, but we decided to go with the mecanum wheels, since they might be slightly more stable and thus easier on the programming and

pathfinding side. Mecanum wheels also transfer a higher torque than omni wheels, which will be helpful since we might not have very strong motors. [5]

Secondly, we considered the size and weight of the robot. We will try to make it as small and light as possible, to help with maneuverability and not put too much stress on the motors. Next is the material. We will be using lego, since that is the task for this project. It is quite a good choice for our project, since it isn't too heavy (at least if we can design a chassis without too many parts), and it's strong enough for our purposes.

Another thing to consider was the gearbox. Adding a multiple-gear gearbox would be impractical, since each wheel needs to spin independently. So we decided on a fixed gear with gear reducers for each motor to increase the torque. The number of motors has to be 4 (one for each wheel), because for a mecanum wheel robot to be able to move side to side and rotate, you need to be able to individually control each wheel and rotate them in different directions. We didn't choose to add any suspension, because that would be too complicated and unnecessary for our project and purposes.

2) Electronics

Main choice is what will be the main controller, for this project Arduino Uno has been chosen because of price, simplicity of work principle and small size but caused problem, lack of pins that got solved by adding an expander PCF8574N.

For control over engines a sufficient and easy to implement in terms of wiring turned out to be L293D motor controller as it allows to pair engines but control all 4 engines independently.

Obstacles and mapping of surroundings will be based on ultrasound sensors as their wide range of working angle allows them to easily check whole surroundings just with 4 sensors.

Path follow will be done with the use of a pair of infrared sensors that will detect change of brightness of base.

To be able to control the robot manually and wireless module HM-10 MLT-BT05 is used.

3) Software

For software as we use an arduino uno a basic arduino IDE is sufficient to debug and upload codes to arduino. Use of arduino partially forced us to code in bit changed C++ that is used by arduino.

Manual control over the robot will be done by use of android bluetooth application.

2. Computer, spatial and mathematical model

One of the most important considerations for the mechanical section of the robot is determining the required torque to move the wheels. To do this we have to model the rolling resistance of the robot. To do this we can use the following formula:

$F_r = C_{rr} N$, where C_{rr} is the rolling resistance coefficient, and N is the normal force.

We want to calculate the force for each wheel, so that we can determine the required torque from each motor. The normal force on each wheel will be equal to $\frac{M}{4}g$, where M is the total mass of the robot, since there are 4 wheels and to simplify it we can assume the weight is equally distributed on each wheel. So the rolling friction for each wheel will be equal to:

$$F_r = \frac{M}{4} C_{rr} g$$

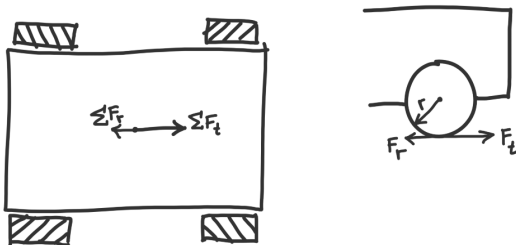


Figure 2

We know that the force the tires exert on the ground should be higher than the force of rolling resistance so that the robot would be able to move. The torque distributes force differently depending on which direction the robot is moving. [5] The force is the smallest for diagonal movement, so we will calculate the necessary torque based on that.

$$F_d = \frac{\sqrt{2}}{2} \frac{\tau}{r}$$

The equation in [5] presents the total force in a 4-wheel vehicle, so the force for each wheel is divided by 4, as above. We know the force of the wheel should be greater than the rolling friction force.

$$\frac{\sqrt{2}\tau}{2r} > \frac{M}{4} C_{rr} g \Rightarrow \tau > \frac{\sqrt{2}}{4} Mr C_{rr} g$$

$$M = 1.64 \text{ kg}, C_{rr} = 0.002 [3, p21], r = 30\text{mm}$$

$$\tau > 0.00034$$

To calculate the required gear ratio, we need to have the electric motor torque. The torque is not specifically listed by the manufacturer, but we can calculate it using the following formula. [6]

$$\tau_m = 9.549 \frac{P}{n}$$

where P is the power in watts and n is the rotational speed in rpm. Using the values from the manufacturer's website (voltage between 3 and 6V, current around 30mA, rpm around 14000), we calculated the torque for the value of minimum voltage to make sure it would work in all cases. The result is:

$$\tau_m = 6 \cdot 10^{-5} \text{Nm}$$

$$\tau = \tau_m \cdot R, \text{ where } R \text{ is the gear ratio.}$$

$$\tau_m R > 0.00034$$

$$R > \frac{0.00034}{6} \cdot 10^5 = 5.67$$

We didn't account for any type of friction or resistance within the mechanism or any other external conditions, and we are not certain about the rolling resistance coefficient, since there were different values listed in different sources, so it would be safer to use a ratio quite a bit above that to make sure it functions. For our model we chose a ratio of 40:1, and used a worm gear to achieve that.

3. Conclusions and comments

Our conclusion is that it's very challenging to imagine and design our systems without the opportunity to actually play with those lego parts in real life.

Part 1	<p>Usage of mecanum wheels serves us well in regard to time thus we also created path recognition and obstacle avoidance software.</p> <p>As a result we were able to put together a working omnidirectional robot.</p>
Part 2	<p>Working in this difficult pandemic reality, our biggest challenge while working on this project was designing with parts that we have never seen before. The possibility of checking our ideas with the use of real parts would significantly speed up the entire design process. Another challenge resulting from the circumstances was working and coordinating this work with people you never met. Our only contact was at Internet conferences which for such an interdisciplinary and complex project introduced a lot of misunderstandings</p>
Part 3	<p>Having a working model of our omnidirectional robot, you should consider what we can use it for.</p> <p>The first thing that comes to mind is an application that has already been proven commercially - cleaning. Fitting brushes or a single brush seems simple and efficient enough. However, the problem may be detecting objects to be swept.</p> <p>Our other idea is to mount an arm that could make sudden jumps in order to mount a cat toy on it. We would use software to make the robot run away from the cat and rotate in place every now and then so that the cat would not get bored too quickly. It is quite simple and there is no competition in the market so far. However, the effectiveness of this idea can only be tested empirically through tests.</p>

4. Bibliography

- [1] Jafari, Moein; Get Rolling with Omni-directional Wheels; (visited on 05.2021);
<https://www.servomagazine.com/magazine/article/get-rolling-with-omni-directional-wheels>
- [2] RoboMaster; Mechanics Of Mecanum Wheel(English version); Theory behind mecanum wheel: (visited in 05.2021);
<https://www.youtube.com/watch?v=Xrc0l4TDnyw>
- [3] R.P.A. van Haendel; supervised by: prof. M. Ang. Jr, Prof. Dr. Ir. M. Steinbuch; Design of an omnidirectional universal mobile platform; non-lego omni wheel robot containing a gearbox: (visited in 05.2021);
<http://www.mate.tue.nl/mate/pdfs/5816.pdf>
- [4] Qian, J. et al; The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System; Sensors 2017, 17(9), 2073;
<https://doi.org/10.3390/s17092073>
- [5] Ether; Mecanum and Omni Kinematic and Force Analysis;
<https://www.chiefdelphi.com/t/paper-mecanum-and-omni-kinematic-and-force-analysis/106153>

[6] The Engineering Toolbox; Electric Motors - Torque vs. Power and rpm;
https://www.engineeringtoolbox.com/electrical-motors-hp-torque-rpm-d_1503.html

5. Appendix A: Software Used:

- Autodesk Fusion 360:

<https://www.autodesk.com/products/fusion-360/overview>

-Arduino IDE

<https://www.arduino.cc/en/software>

-Fritzing

<https://fritzing.org/>