LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

LUT School of Energy Systems

LUT Mechanical Engineering

Weiting Le

DESIGN OF TRACTION TRANSMISSION AND SUSPENSION SYSTEMS FOR AN OMNI-DIRECTIONAL MOBILE ROBOT

Examiners: Professor Heikki Handroos

Ph.D. Ming Li

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Weiting Le

Design of traction transmission and suspension systems for an omni-directional mobile robot

Master's thesis

2015

61 pages, 52 figures, 1 table and 2 appendices

Examiners: Professor Heikki Handroos
Ph.D. Ming Li

Keywords: Mecanum wheel, mobile robot, omni-directional drive, ROS, suspension system, traction system, EPOS controller, SolidWorks.

This project aims to design and manufacture a mobile robot with two Universal Robot UR10 mainly used indoors. In order to obtain omni-directional maneuverability, the mobile robot is constructed with Mecanum wheels.

The Mecanum wheel can move in any direction with a series of rollers attached to itself. These rollers are angled at 45º about the hub's circumference. This type of wheels can be used in both driving and steering with their any-direction property.

This paper is focused on the design of traction system and suspension system, and the velocity control of Mecanum wheels in the close-loop control system. The mechanical design includes selection of bearing housing, couplers which are act as connection between shafts, motor parts, and other needed components. The 3D design software SolidWorks is utilized to assemble all the components in order to get correct tolerance. The driving shaft is designed based on assembled structure via the software as well. The design of suspension system is to compensate the assembly error of Mecanum wheels to guarantee the stability of the robot.

The control system of motor drivers is realized through the Robot Operating System (ROS) on Ubuntu Linux. The purpose of inverse kinematics is to obtain the relationship among the movements of all Mecanum wheels. Via programming and interacting with the computer, the robot could move with required speed and direction.

ACKNOWLEDGEMENTS

This master thesis was carried out during 2015 in the Laboratory of Intelligent Machines, Lappeenranta University of Technology, Finland. This work is a big and challenging project.

I would like to express my sincere gratitude to my supervisor, Dr. Ming Li for giving me many suggestions and encouragement to do the research. He provided me valuable advice and guidance by his wide expertise and experience. These suggestions opened my mind and helped me a lot. I am very grateful to Professor Heikki Handroos for supporting and giving me the opportunity to work in the laboratory. I am also thankful to Dr. Hamid Roozbahani who was the first person I met in the laboratory, he gave me chance to join the team and always encouraging me.

I would like to thank all my team members, especially Mr. Andres Belzunce Tarres. He helped me a lot while doing the ROS programming. Without his help, I could not be done the motor control part well. I appreciate the help and kindness of Professor Huapeng Wu, Mr. Juha Koivisto, Dr. Yongbo Wang and Ms. Bingyan Mao as well.

Deeply love and thanks goes to my parents and friends, without their support and encouragement, the thesis work would not be completed on time.

Weiting Le
Lappeenranta 08.11.2015

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

a	Acceleration of mobile robot [m/s^2]
d	Distance from the center of turning axis to the outside of the wheel [m]
F	Force [N]
F_a	Force produced by acceleration [N]
F_G	Force of gravity [N]
F_H	Slope direction force produced by slope angle [N]
g	Acceleration of gravity [m/s^2]
i	Gear reduction
I	Current [A]
l	Y axis distance form each wheel to the center of gravity [m]
L	X axis distance from each wheel to the center of gravity [m]
l_w	Wheel width[m]
L_r	Roller length [m]
m	Mass of the robot [kg]
n	Number of the rollers
n_1, n_2	Motor speed [RPM]
P	The performance of motor (power) [watt]
R	Radius of the external wheel [m]
S_t	Service factor temperature
T	Torque of the wheel [Nm]
T_2	Output torque [Nm]
T_{KN}	Rated torque of coupling [Nm]
T_N	Rated torque of machine [Nm]
v	Robot speed [m/s]
V	voltage [V]
V_{ir}	Tangential velocity vector of the free roller touching the ground [m/s]
V_{iw}	Velocity vector corresponding to wheel revolutions [m/s]
v_r	Resultant velocity of omni-directional platform [m/s]
x	Value on X axis

y	Value on Y axis
α	Slope ground angle [º]
β	Direction of motion [º]
ω	Angular velocity [RPM]
ω_{iw}	Angular velocity of the wheel [RPM]
φ	Angle of vertex of the roller length projected to the original point [º]
γ	Angle between roller and hub axes [º]
CAN	Controller Area Network
EPOS	Easy Positioning System
ROS	Robot Operating System
RPM	Revolutions per Minute
UR 10	Universal Robot 10
USB	Universal Serial Bus

1 INTRODUCTION

“Robota” is an original expression of robot which was used to denote fictional humanoid. Nowadays, people give “robot” a simple definition that is, according to Introduction to Robots and Robotics (2010) “Robot is a combination of electronics, mechanics and programming (non-programmable in some cases), which senses it is surrounding through its sensors; processes the sensor information and does something in response.”

Mobile robot could not be moved and carried in the past because it was controlled by heavy computer system with many cables and other wire devices (Bräunl, 2006, p. 3). Today, mobile robot is developing that it has an obvious capability to move around instead of being fixed to a location. It can be controlled by light and embedded computer system and wireless devices. Hardware and software aspects are necessary to be considered when building a mobile robot. In the hardware components, locomotion, sensing, reasoning and communication are the four subsystems needed to be taken in account. (Dudek & Jenkin, 2000, p. 15.)

1.1 Problem

Recently, mobile robot topic becomes popular and developing. They are widely used for many applications and occasions, like transportation tasks, house cleaning, and reparation. Sometimes, it appears that when the robot is turning around, especially the wheeled robot, there is an obstacle right on the way. Regardless the driving direction changes of the mobile robot in such case, it always touches the obstacle due to the existence of turning radius of the robot wheels, when the obstacle is happened to be located inside the range of radius. In order to avoid this situation and also let the robot find a new way without the contact with the obstacle to the next step, the robot is supposed to have the omni-directional maneuverability. A great variety of sensors are also needed.

Regarding the large payload of the mobile robot due to the equipped two industrial arms, the mobile platform of the robot needs to have the considerable traction capability, and the big output torque of the driving wheels as well as the good positioning performance are also needed to be taken into account.

1.2 Solution

A mobile robot has been built in the Laboratory of Intelligent Machines. The project is divided into six subtasks that include the traction and suspension system design, carriage chassis design and analysis, control system design of mobile robot, manipulator control and design, quad copter integration, and machine vision development of mobile robot. In this paper, it mainly presents the traction and suspension system design and one part of control system design, the control of motor drivers.

The mobile robot is equipped with four Mecanum wheels, in order to obtain the omni-directional maneuverability. Mecanum wheel has several rollers depending on its surface and can move in any direction which could avoid the situation of turning radius and obstacle. Each wheel is controlled by individual motor, and then using CAN Bus (Controller Area Network open Bus) as network connection to connect all the motors together to the computer. EPOS (Easy Positioning System) controller is applied in the close-loop system to control motors via ROS (Robot Operating System) programming, in order to obtain the desired speed and direction of the mobile robot.

To solve the large payload of the mobile robot and high output torque of the driving wheels, mobile robot platform design that includes the traction system design and the suspension system design introduce in this paper. The traction system design is to improve the capability of traction and meet the output torque which is required. While the suspension system design is to avoid the obstacles and solve the situation when four Mecanum wheels have assembly error. Due to the structure of the Mecanum wheel, it has free rollers. These rollers on each wheel may not contact with the ground at the same time when the wheels rotate. In order to make the mobile robot move smoothly, the suspension system would compensate the contacting errors between Mecanum wheels and the ground.

In this report, design of the traction system including the wheel configuration, hardware selection and kinematics analysis presents in Chapter 2. With constructing a mathematical model of the platform, inverse kinematics is chiefly computed and realized the pose control in the Cartesian coordinate system. The control of motors states via using ROS which is a set of software libraries and tools on Ubuntu Linux to build the robot application. The suspension system design is the main part at the beginning of this project and taken long time to design, though this idea is killed finally because of the space limit. The ideas of suspension system explain in Chapter 3.

1.3 Contribution of the work

The ideal model of the mobile robot would be driven by four Mecanum wheels with two UR 10 (Universal Robot 10) manipulators (see the figure 1). In this research, the traction system is built and applied in the design of mobile robot. On account of the traction system is combined with four Mecanum wheels, and each wheel is controlled by individual motor to obtain different speed and direction when it moves. The purpose of inverse kinematics is to obtain the relationship among the movements of all Mecanum wheels. The equations of inverse kinematics have been solved in real time and the pose control of the mobile robot is realized in the Cartesian space. Via programming on ROS and interacting with computer, the robot can move ideally with desired speed and direction and the velocity of four Mecanum wheels can be obtained respectively as well. This result will promote the exploit and application of inverse kinematics on ROS.

The design of the independent suspension which is set on each wheel can solve the problem of four Mecanum wheels assembly and contacting errors. It can achieve the real-time adjustment of distance between each wheel and ground to compensate the error. In addition, it can also reduce the vibration when the mobile robot bumps into the obstacle. The design of suspension system could be as a reference in the future.

The mobile robot will be applied in the laboratory to do transportation and assembly tasks. It can move in any direction and any speed with its Mecanum wheels. It has 3D vision and variety of sensors to check the surrounding in order to avoid obstacles.

Figure 1. The model of mobile robot.

2 DESIGN OF TRACTION SYSTEM

The design of traction system in this chapter mainly introduces about Mecanum wheel and hardware integration. The purpose of Mecanum wheel selection is to improve the maneuverability and mobility. There are individual motor and driver controlling each wheel to get independent velocity. In order to make the robot move in various direction with any speed, the inverse kinematics is required. The velocity control of individual wheel is controlled realized in the close-loop control system via ROS on Ubuntu Linux. Kinematics of the traction system explains the pose control of the platform realized in the Cartesian space.

2.1 Wheels

The design of mobile robot is needed to consider three fundamental characteristics which is maneuverability, controllability and stability based on both wheel types and configuration. As mentioned before, omnidirectional wheels are selected because it has better maneuverability and mobility to move in any direction. This part introduces some information about Mecanum wheel and the mechanical design of this type of wheel.

2.1.1 Type and configuration

Different wheel classes for a mobile robot has different effect on their locomotion mechanism. The choice of wheel types depends on wheel arrangement, or wheel geometry (Goris, 2004, p. 11). Both wheel type and wheel configuration can influence three basic properties of a robot, maneuverability, controllability and stability (Goris, 2004, p. 14).

The minimum number of wheels is two, however static stability requires a minimum of three wheels under ordinary circumstances. It is possible to add more wheels to improve the stability while designing the wheeled robot. Like omni-directional robots, they can move in any direction instantly without changing the orientation unlike other designs. This shows the maneuverability of the robot. (Doroftei, Grosu & Spinu, 2007, p. 512.)

Omni-directional wheels have three degrees of freedom and can move in any direction. There are small passive rollers which provide low resistance around the circumference. Omni-directional wheels are called Swedish wheels as well because this wheel class has been designed by Swedish company. One of the commonly used omni-directional wheel is named Mecanum wheel, which is developed and patented by the Swedish company Mecanum AB (Bräunl, 2006, p. 113). The angle between rollers axis and central wheel typically could be 45°.

Mecanum wheels can be used in both driving and steering with their any-direction property. However, they still come with several drawbacks such as poor efficiency, high price and positional control problem.

There are 4 Mecanum wheels with individual motor on each wheel. It is more efficient if compared to 3 wheel configuration which is suitable in this case. When the robot moves in one direction, two wheels can move at the same orientation of motion that makes robot drive with enough speed. (Goris, 2004, p. 26.)

Omni-directional drive can move and turn in any direction smoothly, and have capability of precise movements. In spite of Mecanum wheel costs a bit higher and has a challenge of controllability, it has a perfect maneuverability, high load capacity and acts well in its application.

2.1.2 Design of Mecanum wheel

The wheel (see the figure 2) consists of a hub which carries a number of free moving rollers, and these rollers are angled at 45° about the hub's circumference. (Doroftei et al., 2007, p. 517.)

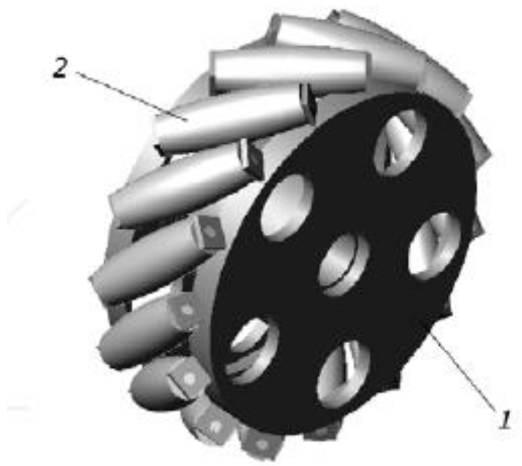


Figure 2. Mecanum wheel, where 1 is hub, and 2 is roller (Doroftei et al., 2007, p. 517).

The rollers are shaped as in the figure 3 to obtain the contour of the omni-directional wheel is circular. The angle γ = 45º is selected, which is between roller and hub axes (see the figure 3). Each roller can be a finite length because the axis of rotation is offset by the angle γ to the axis of the wheel. The profile of the roller would be the arc of an ellipse (Doroftei et al., 2007, p. 518). The roller shape should respect the equation:

$$\frac{1}{2}x^2 + y^2 - R^2 = 0 \tag{1}$$

, where R is the radius of the external wheel, x is the value on X axis and y is the value on Y axis. (Doroftei et al., 2007, p. 517-518.)

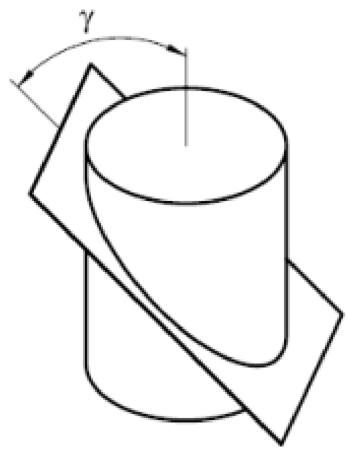


Figure 3. The angle γ (Doroftei & Stirbu, 2010, p. 174).

The roller shape can be seemed as a circle at first, then layout of rollers can be seen in the figure 4 so that getting the circular silhouette for the wheel. The rollers are less needed to span the circumferential area of the wheel, because the axis of rotation becomes closer, each roller will get longer. (Doroftei et al., 2010, p. 174.)

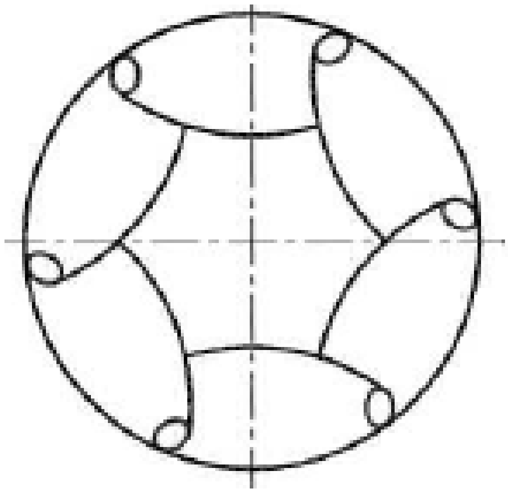


Figure 4. Layout of the rollers (Doroftei et al., 2010, p. 174).

The roller length L_r should respect the equation:

$$L_r = 2R\frac{sin\frac{\varphi}{2}}{sin\,\gamma} = 2R\frac{sin\frac{\pi}{n}}{sin\,\gamma} \tag{2}$$

$$\mathrm{n} = \frac{2\pi}{\varphi} \tag{3}$$

, where R is the radius of the external wheel, n is the number of the rollers (see the figure 5), and

$$\varphi = 2acrsin(\frac{L_r}{2R\sin\gamma}) \tag{4}$$

, where γ is the angle between roller and hub axes, φ is the angle of vertex of the roller length projected to the original point.

The wheel width l_w is the equation (Doroftei et al., 2007, p. 519):

$$l_w = L_r\cos\gamma = 2R\frac{\sin\frac{\pi}{n}}{\tan\gamma} \tag{5}$$

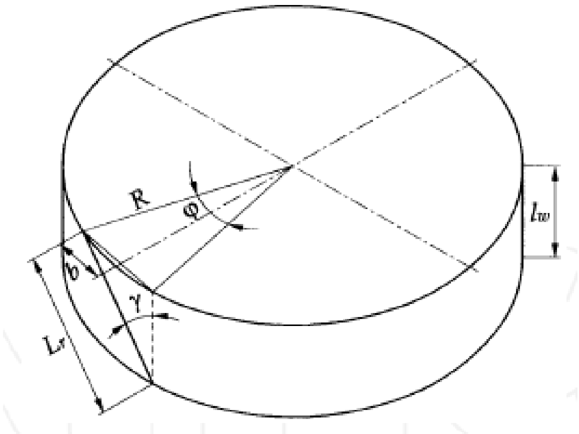


Figure 5. Symbols shown in the wheel part (Doroftei et al., 2007, p. 519).

As γ= 45º is selected, then the equations become (Doroftei et al., 2007, p. 520):

$$L_r = 2\sqrt{2}\mathrm{R}\sin\frac{\pi}{n} \quad (6)$$

$$l_w = 2R\sin\frac{\pi}{n} \quad (7)$$

We assume the radius of the wheel R is equal to the roller length L_r, then we get:

$$\mathrm{n} = 8.69 \approx 8 \quad (8)$$

The number of rollers is 8, and the maximum roller length is the same as the radius of the wheel.

According to the dimensions of platform with 1600 mm length, the diameter of the wheel R is considered around 300 mm to make the whole appear attractive. In order to obtain good performance of the wheel, the whole Mecanum wheel set is bought from market. After going through some websites, a set of 254 mm Mecanum wheel (4 pieces) from Nexusrobot online market is as an option. There are two types of shape to be selected, which are mainly made of aluminum block and plate respectively (plate type Mecanum wheel see the figure 6) (Luo, 2015). Considering the robot in this case will carry around 250 kg weight, so the main material which is aluminum block would be a better choice to make wheel have more payload. Thus, 254 mm Aluminum Mecanum wheels (see the figure 7) is selected. The parameters shows in the following table (as table 1).

Table 1. The specification of Mecanum wheel (Luo, 2015).

Diameter	254 mm
Width	130 mm
Number of Rollers	8
Body Materials	Aluminum alloy
Roller Materials	Polyurethane
Spacer Material	Aluminum
Length of Roller	92 mm
Diameter of Roller	38.2 mm
Net Weight	7.0 kg * 4
Load Capacity	> 400 kg

Figure 6. Mainly made of aluminum plate (254mm Mecanum Wheels with PU roller, 2015).

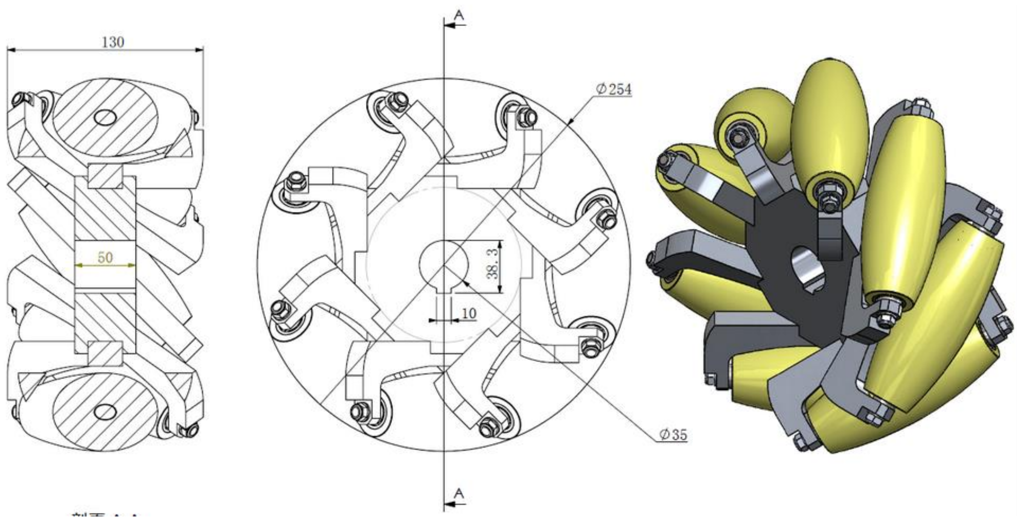


Figure 7. The 254 mm Aluminum Mecanum Wheel (Luo, 2015).

2.2 Locomotion

Each Mecanum wheel is driven individually by the electric motor. The motor selection is made on the basis of computation of the general torque and velocity that is a linear function of the acceleration and mass of the mobile robot. The selection of the controller and drive system of the motor depends on the particular performance requirement of the motor, and each motor is individually controlled in order to obtain the independent velocity control in each wheel. The mechanical design of the traction system is carried out in the SolidWorks.

2.2.1 Motor selection

The servo motor integrated with the gearhead is applied here, and the traction system benefits from its compact size. Both the torque and velocity of the motor need to be controlled in the traction system.

- Motor torque calculation

The output torque of the motor is the quantity of force where its output shaft can rotate. The output of the motor shaft may stall, or even stop turning if torque is over applied to it. Torque is relating to force and distance, which is:

$$\mathrm{T} = \mathrm{F} \times \mathrm{d} \tag{9}$$

$$\mathrm{F} = \mathrm{m} \times \mathrm{a} \tag{10}$$

, where T is torque, F is the force, and d is the distance which from the center of the turning axis to the outside of the wheel is the radius of the wheel R, m is the mass of robot, and a is the acceleration. (Neal, 2010, p. 34-35.)

As Mecanum wheel with diameter 254 mm (R = 0.127 m) is chosen, and the robot acceleration is assumed to 1 m/s^2, 250 kg as general whole robot weight. There are two situations needed to think about, one is flat ground, and the other is slope one (see the figure 8). Considering the robot is mainly applied in the laboratory, when it moves in any case, some screws on the ground would be an obstacle. Thus, the angle of slope surface with ground is supposed to be $\alpha = 10°$. According to equation (9) (10), the following is about the related calculations.

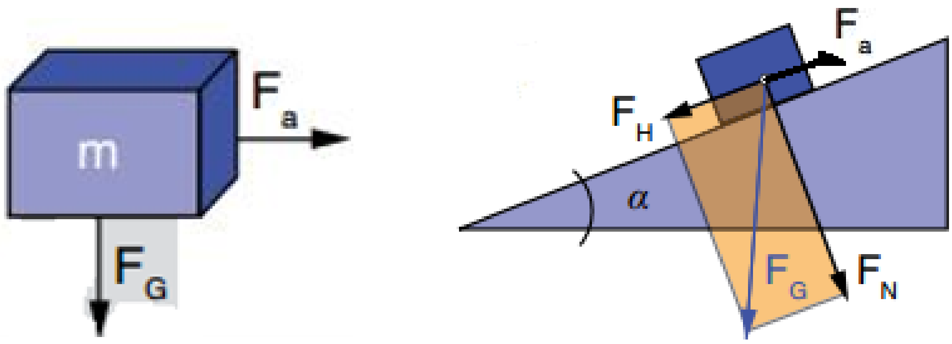


Figure 8. Left figure is about flat ground situation and right one if about slope one.

One situation (flat ground):

$$F_G = m \cdot g = 250\ kg\ \cdot 10m/s^2 = 2500\ N \tag{11}$$

$$F_a = m \cdot a = 250\ kg \cdot 1m/s^2 = 250\ N \tag{12}$$

$$T = F_a \cdot R = 250\ N \cdot 0.127\ m = 31.75\ N \cdot m \tag{13}$$

, where F_G is the force of gravity, g is the acceleration of gravity in this situation where the value is 10 m/s^2. F_a means the force produced by the assumed acceleration a. T is the whole torque of the four wheels. Torque of each wheel on the flat ground:

$$T_i = \frac{1}{4} \cdot T = 7.94\ N \cdot m \tag{14}$$

Another situation (slope ground α = 10°):

$$F_G = m \cdot g = 250\ kg\ \cdot 10m/s^2 = 2500\ N \tag{15}$$

$$F_H = F_G \cdot \sin\alpha = 2500\ N \cdot \sin 10° = 434.12\ N \tag{16}$$

$$F_a = m \cdot a\ +\ F_H = 250\ N + 434.12\ N = 684.12\ N \tag{17}$$

$$T = F_a \cdot R = 684.12\ N \cdot 0.127\ m = 86.88\ N \cdot m \tag{18}$$

, where F_H is the slope direction force produced by slope angle. Torque of each wheel on the slope ground:

$$T_i = \frac{1}{4} \cdot T = 21.72\ N \cdot m \tag{19}$$

Learnt from these two results that is the maximum torque of each wheel would be 21.72 N · m for this moment.

- Motor speed calculation

The output speed of the motor is the rotating velocity where the output shaft revolves. There is the output speed reducing when the payload of the motor rises.

Speed in motor is proportional to the voltage applied to the rotor. The following equations show the relationship between the power produced by the motor P, voltage V and current I. (Neal, 2010, p. 33.)

$$P = V \times I \tag{20}$$

$$P = T \times \omega \tag{21}$$

, where T is torque, ω is angular velocity in radians per second. It can increase the voltage rating or current to increase the power output, but the more voltage rating increases, the more dangerous the operator will be. Thus, in this case, 24 or 48 volts motor which is commonly used for medium mobile robot is preferred to use. As seen from equation (21), the motor selection is also mainly based on rotation speed and torque. Motor torque turns the robot wheels and drives the robot. And the robot speed depends on the rotation speed of the motor shaft and the diameter of the wheels. There would be two situations to be considered, one is that if the wheel is directly mounted on the motor shaft, and the other is the wheel is not attached directly to the shaft, but with additional gearing first (see the figure 9). (Neal, 2010, p. 33-34.)

Changing Gear Ratios

If the wheel is attached directly to the gear head motor shaft,
then the wheel and motor angular speeds are the same.

If you add additional gearing,
then
wheel speed = motor speed / G
wheel torque = G x motor torque
G = gear ratio
= output teeth / input teeth

Larger output gear gives slower more "torquey" robot

Figure 9. Two situations about the relationship of speed and torque between wheel and motor (Piccirillo, 2009, p. 6).

The speed of robot assumed would be around 2.5 m/s, considering the efficiency loss of each roller produced, the speed calculated would be higher than 2.5 m/s. According to equations about angular to linear velocity and unit conversion:

$$v = R \cdot \omega \cdot reduction \tag{22}$$

$$1\,rpm = 2\pi/60\,rad/s \tag{23}$$

, where v is linear velocity, ω is angular velocity in radians per minute (RPM), and R is the radius of Mecanum wheel.

Results:

$$2.5m/s = 0.127\,m \cdot \omega \cdot reduction \tag{24}$$

$$\omega \cdot reduction = 19.685 \tag{25}$$

$$rpm \cdot \left(\frac{2\pi}{60}\right) \cdot reduction = 19.685 \tag{26}$$

$$rpm \cdot reduction = 187.978\,rpm \tag{27}$$

The relationship with angular velocity and reduction of gearhead is 187.978 rpm.

After visiting professional and worldwide motor websites, EC 60 Brushless, 400 Watt motor with Hall sensors (see the figure 10) from Maxon Motor which is the leading provider of high-precision drives is selected. As shown in the figure, the nominal speed is 4940 rpm and nominal torque is 810 mNm, if the reduction of gearhead is 25 which will state in the late part, then both torque and speed would meet our requirement.

Nominal voltage	48 V
No load speed	5380 rpm
No load current	449 mA
Nominal speed	4940 rpm
Nominal torque (max. continuous torque)	810 mNm
Nominal current (max. continuous current)	9.84 A
Stall torque	11800 mNm
Stall current	139 A
Max. efficiency	89 %

Figure 10. EC 60 Brushless Motor and its specification (Maxon Motor product, 2015).

Results:

$$v = R \cdot \omega \cdot reduction = 0.127\ m \cdot 4940\ rpm \cdot \left(\frac{2\pi}{60}\right) \cdot \left(\frac{1}{25}\right) = 2.64 m/s \qquad (28)$$

As a result, the robot will get velocity about 2.64 m/s.

2.2.2 EPOS controller

The purpose of building the robot is to have better maneuverability and mobility working in the laboratory. Using the Mecanum wheel makes individual motor control each wheel. The controllability of motor affects performance of wheel, and programming makes motor obtain reasonable rotation velocity with desired command. The controllers are chosen from Maxon Motor as well. EPOS is a modular, digital positioning controller with CANopen (Controller Area Network open). Due to the specification of the motor, the power is 400 Watt. Thus, EPOS2 70/10 Digital positioning controller meets our requirements which has up to 700 Watt power (see the figure 11). (EPOS, 2015.)

EPOS2 70/10
- DC and EC motors up to 700 W
- Point to point control unit (1 axis)
- Interpolated Position Mode (PVT)
- Combination of several drives via CAN Bus
- CANopen
- 10 digital inputs
- 5 digital outputs
- 2 analog inputs
- Robust design

Slave version (online commanding) using CAN Master (EPOS2 P, PC, PLC, SoftPLC, etc.) or PC via USB or RS232 interface

Typical applications:
- Production equipment
- System automation tasks
- Plant construction

Figure 11. EPOS2 70/10, Digital positioning controller (Maxon Motor product, 2015).

Each motor is connected with each EPOS controller first, according to the guide from Maxon Motor (see the figure 12). After that using CAN-CAN cable to connect 3 pairs, the rest of them is communicated to the computer via USB (Universal Serial Bus) cable. EPOS controller is easy to program with C language or C ++. All the motor driver are connected and programmed on Ubuntu system in later chapter.

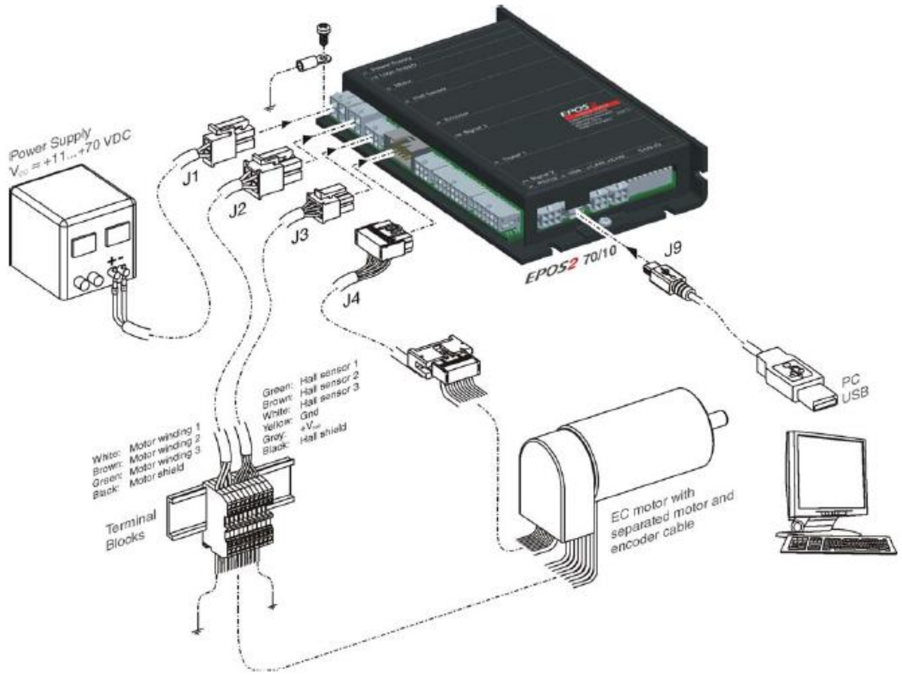


Figure 12. Maxon EC motor (Maxon EC motor, 2015, p. 14).

2.2.3 Hardware of traction control

As motor is selected from Maxon Motor Company, the related components are needed to be decided as well. The specification of gearbox and motor controller is dealt with this part. Accessories such as bearing, couplers and keys present via figures. After that, all of them are assembled together via SolidWorks software in the next part.

• Gearbox and brake

As mentioned before, the torque of motor should be at least 18-20 Nm. However, it is a bit difficult to get such kind of motor used for the mobile robot. There is a rule that when revolution goes high, the torque goes down, vice versa. Thus, the gearbox is considered to be applied to amplify the output torque to meet our desired torque.

Normally, there are two types of gear motor, spur and planetary gear trains. Spur gearheads may lead to fracture if the teeth of gear is repeatedly stressed. It does not apply the torque to the actuation object as well. Considering these factors, the planetary gear motor is chosen though it has higher cost, more complex design. However, it is easy to get very low gearings in a small space, higher efficiency and lower noise level which is good for our case.

Due to the decision of the motor, the normal torque of the motor is around 810 mNm. With configuration option in the official recommendation, Planetary Gearhead GP 81 A (see the figure 13) is selected. The reduction is 25:1. Like previous calculation about motor speed, this reduction gives the robot around 2.64 m/s speed. Then the final torque for each wheel is around 20.25 Nm which meets the requirement. Brake and sensor are also selected during the combination options. The motor products are finally assembled together while delivered to us (see the figure 14). As mentioned, four of them are required.

General information

	Gearhead type	GP
	Outer diameter	81 mm
	Version	A

Gearhead Data

	Reduction	25 : 1
	Absolute reduction	1701/68
	Max. motor shaft diameter	14 mm
	Number of stages	2
	Max. continuous torque	60 Nm
	Max. intermittent torque	90 Nm
	Direction of rotation, drive to output	=
	Max. efficiency	75 %
	Average backlash no load	0.6 °
	Mass inertia	125 gcm²
	Gearhead length (L1)	113.7 mm
	Max. transmittable power (continuous)	750 W
	Max. transmittable power (intermittent)	1100 W

Figure 13. Planetary Gearhead GP 81 A and its specification (Maxon Motor product, 2015).

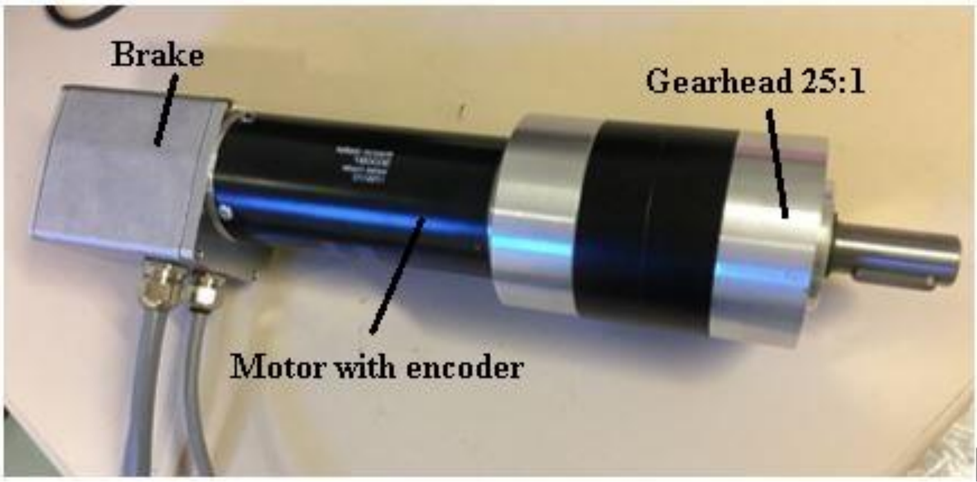


Figure 14. Motor part.

• Tapered bearing

In order to rotate the wheel and motor part, bearing is needed. Tapered roller bearing as main bearing in the transmission part is selected because this kind of bearing can carry both large axial forces and radial forces which is pretty suitable in this case. As assumed before, the overall weight is 250 kg. If the permissible thrust load of the bearing is above 2500 N, then it would hold and work well. The following figure 15 is Type E housed unit bearing which is from Timken Company, the maximum thrust load is 11520 N and allowable slip fit radial load is 22241 N which is totally meet the requirements. (Tapered Roller Bearings, 2015.)

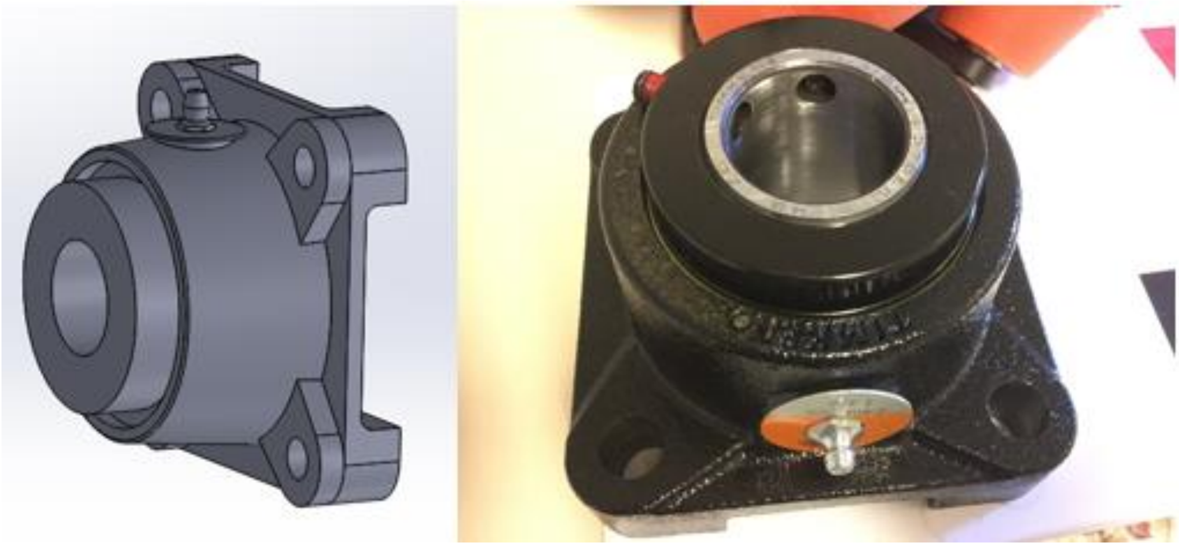


Figure 15. Timken Tapered Bearing.

• Couplers and shaft

Due to the different diameter of gearhead and drive shaft, we have to select suitable coupler as connection. The uneven process of the number of prime movers may cause the damping torsional vibrations and absorbing shocks. The purpose of the couple is utilized to transmit the torque. (Rotex coupler, 2015.)

Based on the specification of motor, the motor torque is 810 mNm, rotation speed is 4940 rpm (no load speed 5380 rpm). According to the equations from Coupler KTR Company:

$$T_{KN} \geq T_N \cdot S_t \tag{29}$$

, where T_{KN} is the rated torque of coupling, T_N is the rated torque of machine (here is 0.810 Nm·25=20.25 Nm), and S_t is the service factor temperature (see the figure 16, we assume the temperature would be above 120 ℃). (Rotex coupler, 2015.)

Result:

$$T_{KN} \geq 20.25\ Nm \cdot 3 = 60.75\ Nm \tag{30}$$

Temperature factor S_t											
	-50 °C	-30 °C +30 °C	+40 °C	+50 °C	+60 °C	+70 °C	+80 °C	+90 °C	+100 °C	+110 °C	+120 °C
T-PUR®	1,0	1,0	1,1	1,2	1,3	1,45	1,6	1,8	2,1	2,5	3,0
PUR	–	1,0	1,2	1,3	1,4	1,55	1,8	2,2	–	–	–

Figure 16. Temperature factor St table (Rotex coupler, 2015).

As mentioned before, the robot is planned to move on the flat in the most of time, so 60.75 Nm could be seen as instantaneous value. Following the result and the figure 17 below, ROTEX size 24 is selected which has rated torque 60 Nm.

ROTEX® size	Max. speed		Twist angle ϕ with		Torque [Nm]		
	V=35 m/s iron	V=40 m/s steel	T_{KN}	$T_{K\,max}$	Rated (T_{KN})	Max ($T_{K\,max}$)	Vibratory (T_{KW})
14	22200	25400	6,4°	10°	12,5	25	3,3
19	16700	19000			17	34	4,4
24	12100	13800			60	120	16
28	10100	11500			160	320	42
38	8300	9500			325	650	85
42	7000	8000			450	900	117
48	6350	7250			525	1050	137

Figure 17. ROTEX size table (Rotex coupler, 2015).

Due to the different diameters of motor shaft and wheel shaft, two types of couplers are selected (see the figure 18).

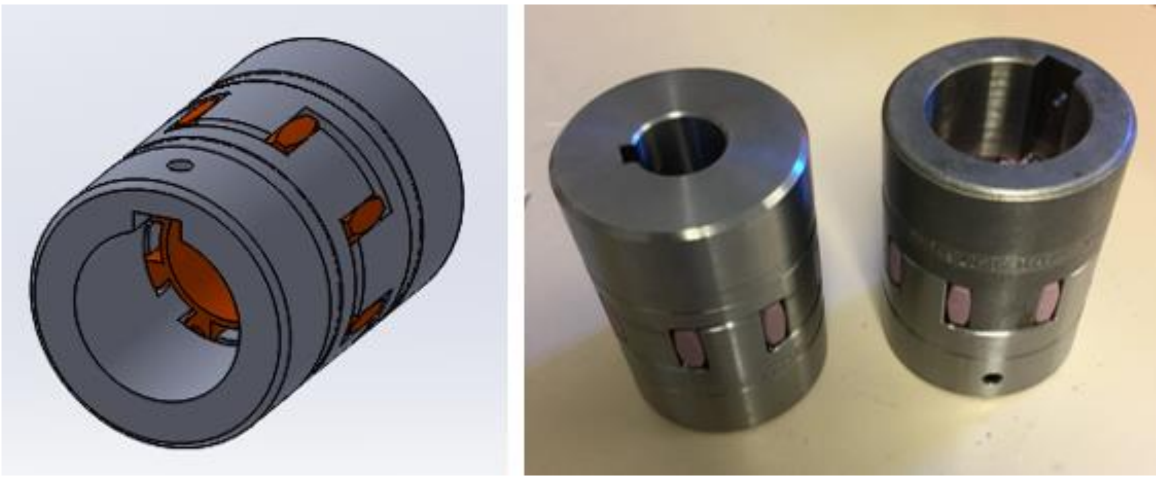


Figure 18. Couplers.

According to the drawing of Mecanum wheel, the diameter of bore is 35 mm. The drive shaft manufactured is with screw thread which is used to fix the wheel outside surface with lock nut and two keyway as connections (see the figure 19 and appendix 1). In order to get good performance with lock nut of tapered bearing,

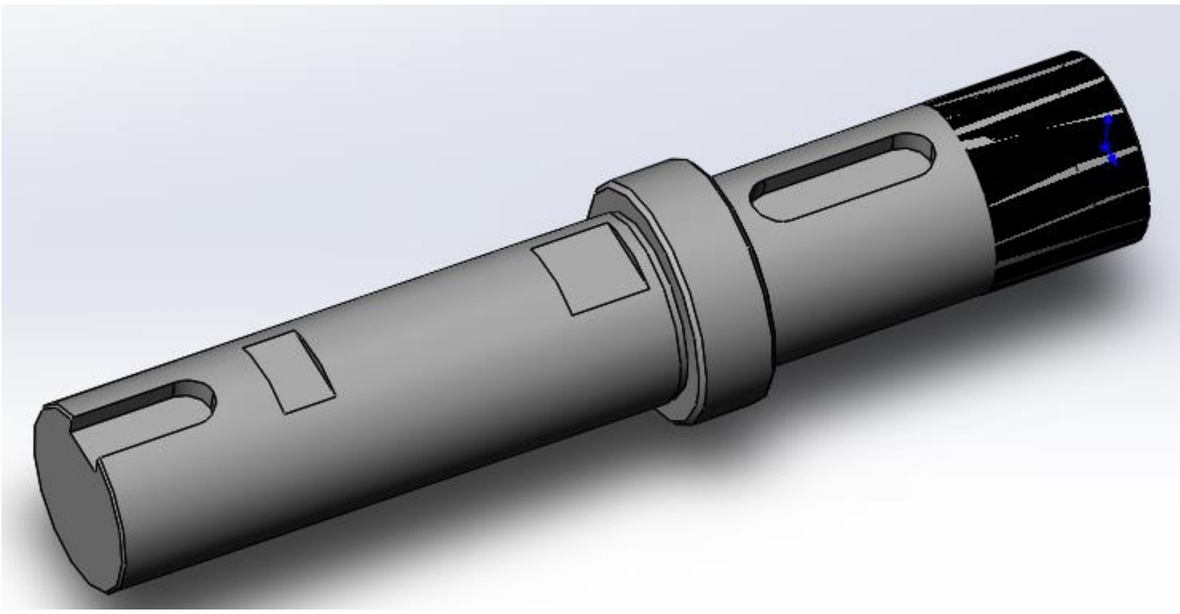


Figure 19. Manufactured shaft.

• Bevel gearbox

Bevel gears are gears utilized to fit the purpose of transmission laid out in the certain angle setting. Most of the bevel gears are mounted on shafts that are right angle apart. In order to remain the torque of wheel unchanged, the reduction of the bevel gearbox is better to be selected 1:1. The equation about output torque is:

$$T_2 = 9550 \cdot \frac{P}{n_2} \quad (31)$$

$$i = \frac{n_1}{n_2} \quad (32)$$

, where P is performance at n_1 here is motor output 0.4 kW, n_1 is motor speed, i is reduction, and T_2 is output torque.

Due to i = 1, n_2 is equal to n_1 which is 4960 rpm. After calculation, we got T_2 is 0.77 Nm. The reduction of gearhead is 25, so the final output torque would be around 20 Nm. After searching in the figure 20, the power gear (see the figure 21) is selected which is from Graessner Company. The output torque of the power gear is 45 $N \cdot m$ which is meet the calculation. (PowerGear, 2015.)

	Abbr.	Unit	P54	P75	P90	P110
i=1:1	T_{2N}	Nm	15	45	78	150
	T_{2B}	Nm	23	68	117	225
	T_{2Not}	Nm	30	90	156	300
i=1.5:1	T_{2N}	Nm	15	45	78	150
	T_{2B}	Nm	23	68	117	225
	T_{2Not}	Nm	30	90	156	300
i=2:1	T_{2N}	Nm	12	42	68	150
	T_{2B}	Nm	18	63	102	225
	T_{2Not}	Nm	24	84	136	300

Figure 20. Power gear performance table (PowerGear, 2015, p. 6).

Figure 21. Bevel gearbox from Graessner Company.

• Power of robot

Based on the specification of motor and robot hand we selected, 48 V batteries is an option. GWL/Power WB-LYP60AHA Life YPO4 (see the figure 22) is a special series of high power lithium cell. Each cell has 3.2 V nominal voltage, so 16 pieces of them are required.

Specification

Weight (kg)	2.3
Height (mm)	203
Width (mm)	114
Depth (mm)	61
Nominal Voltage (V)	3,2
Capacity (Ah)	60
Max discharge current (A)	600
Optimal discharge current (A)	30
Max charging current (A)	180
Optimal charging current (A)	30

Figure 22. GWL/Power WB-LYP60AHA Life YPO4 battery and its specification (mod. Winston Battery, 1994).

According to the relationship between Amp-hours and Watt-hours, the power would be totally 1600 watts and 48 volts battery, then:

$$\text{Amp-hours} = \text{Watt-hours} / \text{volts} \tag{33}$$

$$60\ \text{Ah} = \text{Watt-hours} / 48 \tag{34}$$

The results is Watt-hours is 2880, and hours is 1.8 h. Thus, the robot will run around 1.8 hours in this case.

• Other components

According to the manufactured shaft drawing, there are two types of keys needed which are one side round type and both side round type acting as connection between the Mecanum wheel and coupler (see the figure 23).

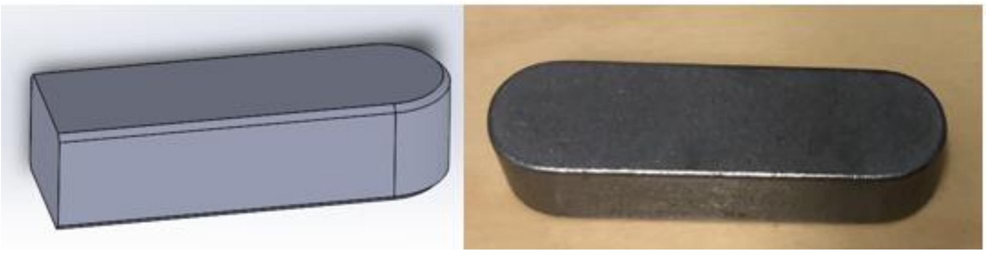


Figure 23. 10-35 Keys.

According to Twin FU Nut (2008) "The locknut selected has two friction rings and a spacer which are secured to the upper surface of the nut with excellent locking performance. The two friction rings are arranged so that stress P generated by the spring action and the reaction P' act symmetrically about the shaft center. This arrangement ensures an even contact force F around the contact face" (see the figure 24). It has enough dead load to avoid wheel rotating axially.

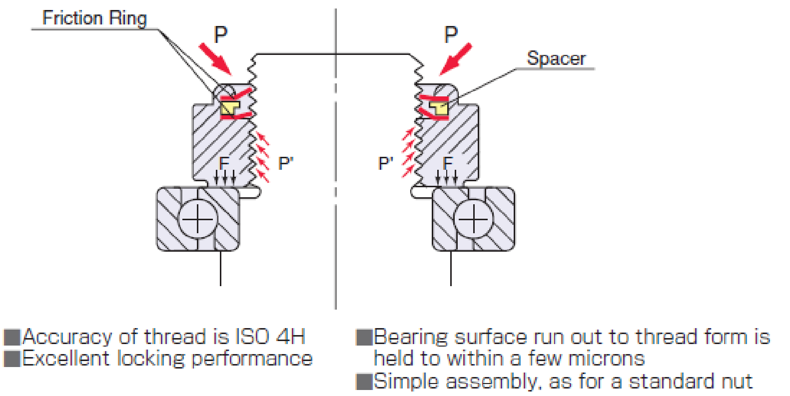


Figure 24. The analysis of the locknut (Twin FU Nut, 2008).

2.2.4 3D model of traction system

The general dimensions of mobile robot is represented in the following image (see the figure 25). The exact size of the robot is unknown, the design is refined step by step according to later requirements and calculations.

General Dimensions	mm
Length	1600
Width	800
Height	unknown

Figure 25. General dimension box.

After assembling them in SolidWorks, the general layout has difference with the first conception. All the transmission components are shown as the translational configuration which made whole chassis look like square (see the figure 26). The length of the platform was 1300 mm and the width was 1000 mm. Thus, another option is to set the motor part by using extra bevel gearbox to change the direction of the motor parts.

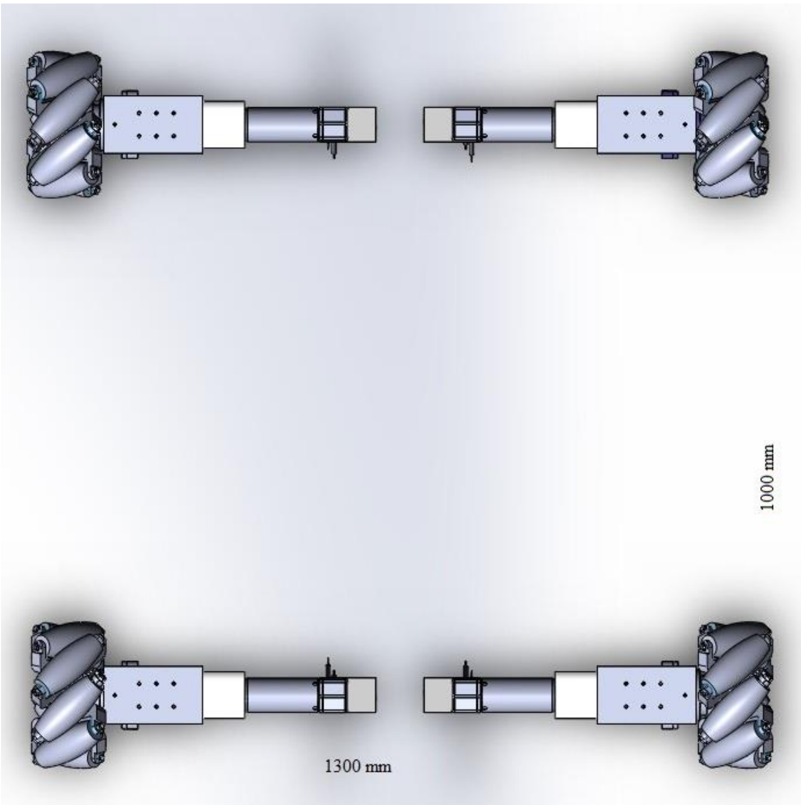


Figure 26. First idea of the transmission part layout.

The following image is the new layout of the transmission part which makes the whole platform closer than expected one (see the figure 27). The length becomes to 1525 mm and the width is 800 mm. Figure 28 shows the details of the transmission system which are assembled via SolidWorks.

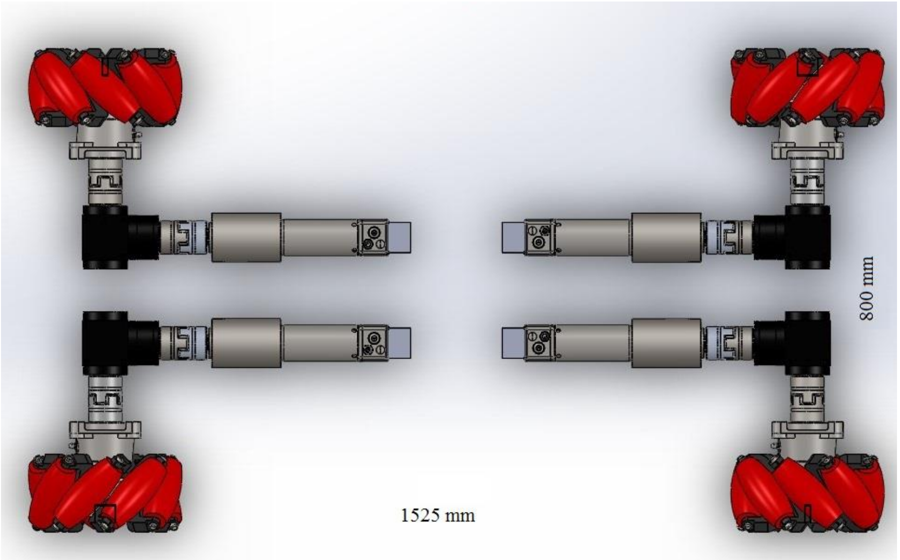


Figure 27. New layout of the transmission part.

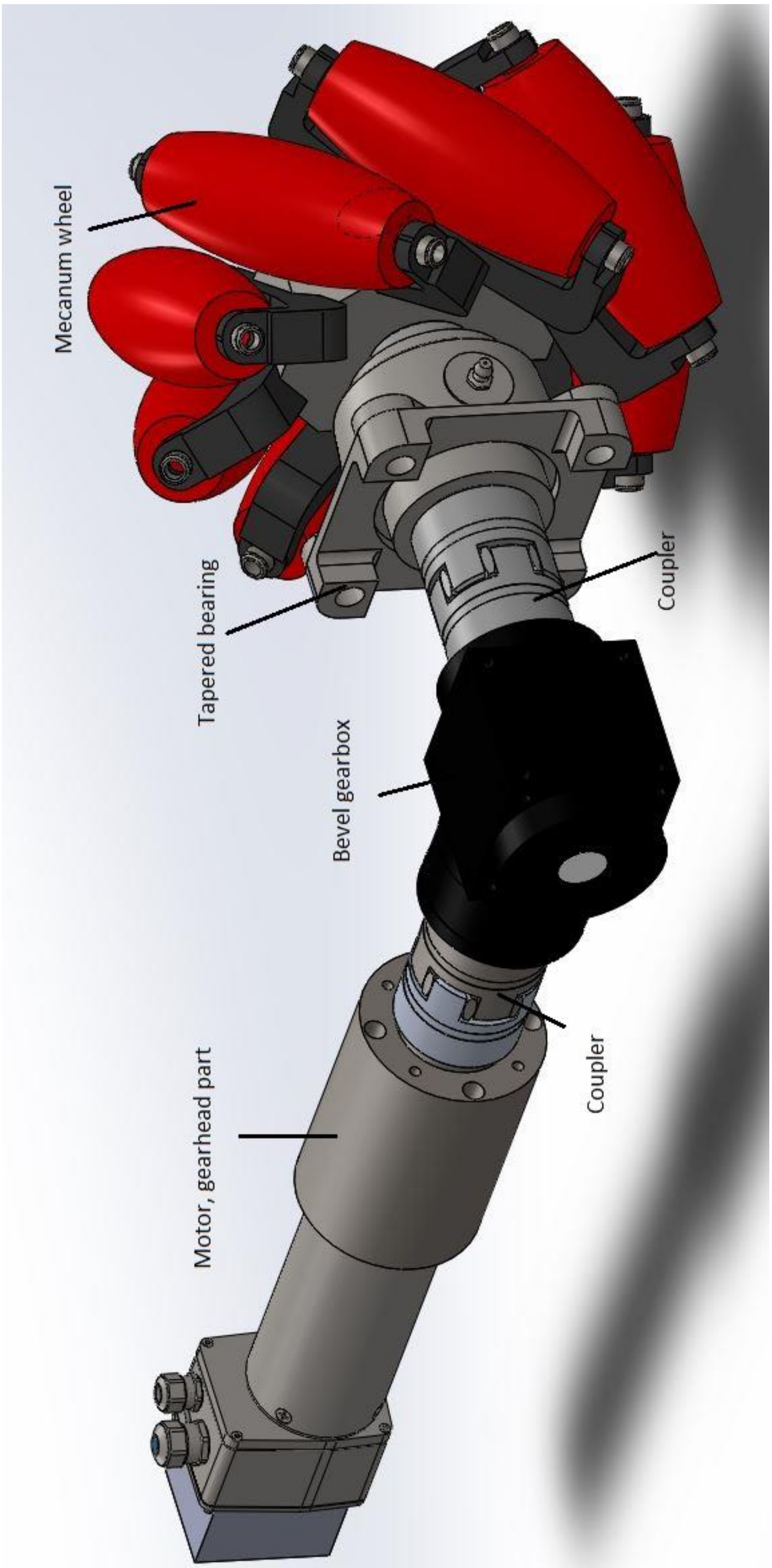


Figure 28. The transmission system assembled via SolidWorks.

An omni-directional drive with four Mecanum wheels is taken in order to get high stability and maneuverability. Each wheel has three degrees of freedom and its own motor for independent drive and steer. These wheels are attached to the robot body, and two wheels on each side of the chassis. The wheel with +45º roller and wheel with -45º roller on each side. Due to the rollers, it would create forces when moving. Thus, the wheels of the robot is needed to have opposite orientation. (Wakchaure et al., 2011.)

Otherwise, the robot will not be able to spin well. The general wheels layout can be seen in the following figure 29. From the top view, it is shown like "X" configuration, and "O" shown from the bottom view.

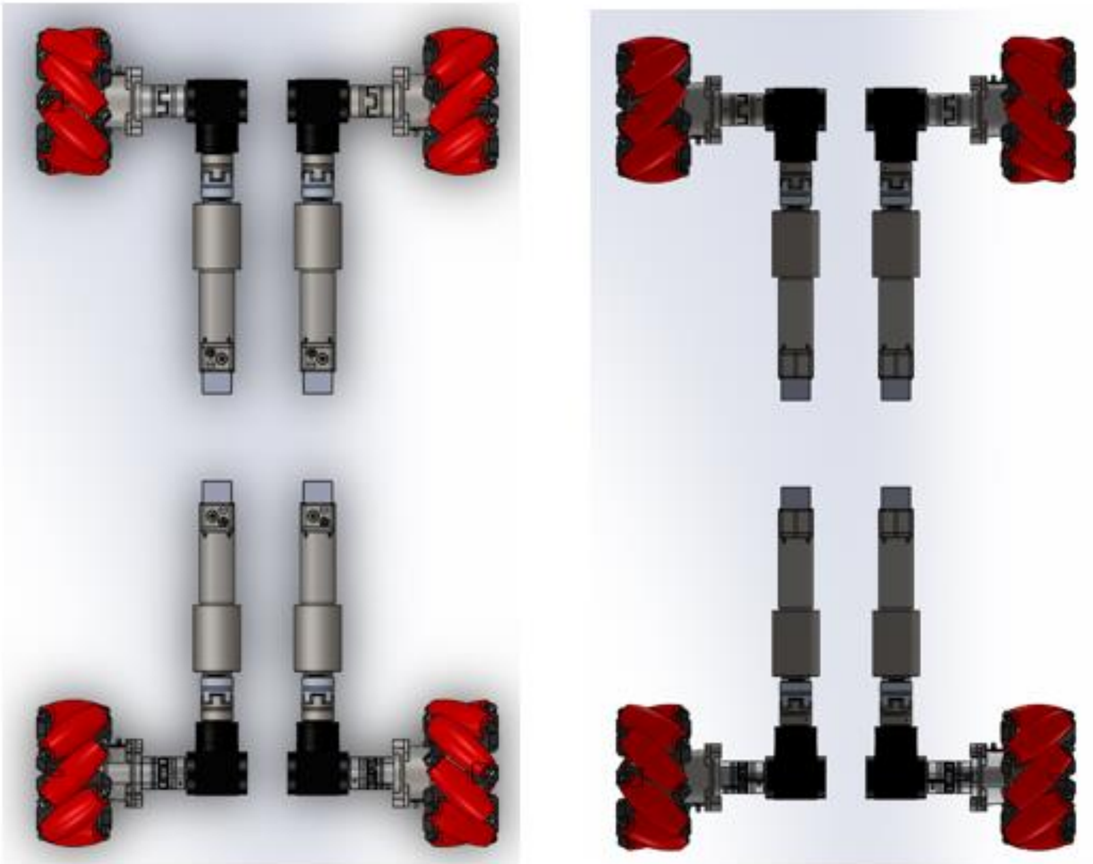


Figure 29. Layout of wheels (left is from top view "X" configuration, right is from bottom view "O" configuration).

2.3 Kinematics of traction system

Kinematics explains robot movements. Kinematic analysis is one of the steps during the robot design. Due to the Mecanum wheel selection, both wheel and rollers of the motions are needed to be taken into account. The purpose of inverse kinematics is to obtain the relationship among the movements of all Mecanum wheels and applied when programming on ROS. Each Mecanum wheel is controlled by motor individually. The idea of control is that each motor is connected to the controller respectively, and then they can be connected in series via CAN bus. It will state in details later in this part.

2.3.1 Inverse kinematics

The maximum loading capacity of whole robot is around 250 kg. Four Mecanum wheels with a diameter of 254 mm and width of 130 mm are provided by Nexus Company. Each of them is driven by gearbox and motor independently. The assumed maximum speed of robot would be 1 m/s with an acceleration time of 2 seconds.

As mentioned before, Mecanum wheel is combined with a certain number of rollers distributed uniformly around the boundary of the wheel. It is actuated by supplying motion through motor, a part of the force is translated by the angled rollers in wheel rotational direction and given as a vertical force in the wheel direction. In order to allow the robot moving in any direction with a total force vector expected, this force is combined with all the forces produced by the velocity of each wheel individually. Due to its situation, there is at least one point of the ground contact by one roller at any moment, slipping becomes a frequent predicament with the Mecanum wheel. However, it can be solved because of the dynamics of this kind of wheel. Even if the Mecanum wheel is driven in the Y direction, the force vectors are still produced in both the X and Y direction (see the figure 30). (Wakchaure et al., 2011.)

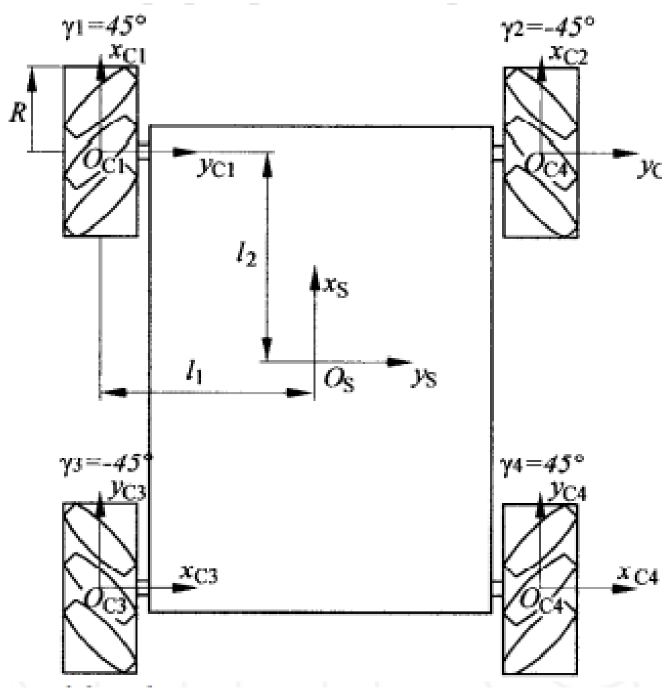
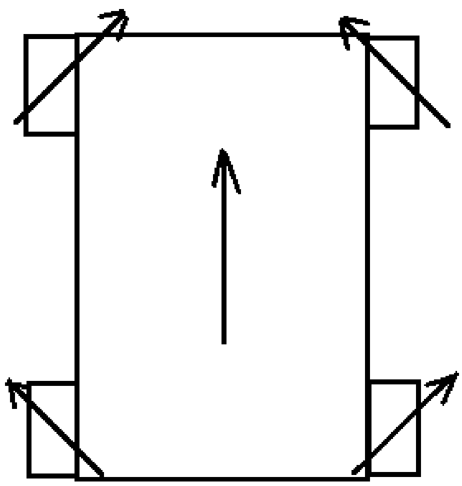


Figure 30. Symbols used on platform (Doroftei et al., 2007, p. 523). The meaning of each symbol will be explained later.

Unlike the normal wheels, Mecanum wheels provide difference force on the ground. The right way to place the wheels is opposite on both side. For example, assuming the robot will move forward, then the creating force is like (see the figure 31):



Moving forward

Figure 31. Different force and direction of the wheel effects the overall robot movement, and this figure is only shown moving forward.

Mecanum wheels produce forces when the robot is moving forward. There is an opposite force created when they are reversed. A force is created by the left side wheels while the right side create an opposite force and then forces offset. Finally the robot moves straight. There are 12 kinds of basic movements shown below with each wheel moving direction (see the figure 32). (Wakchaure et al., 2011.)

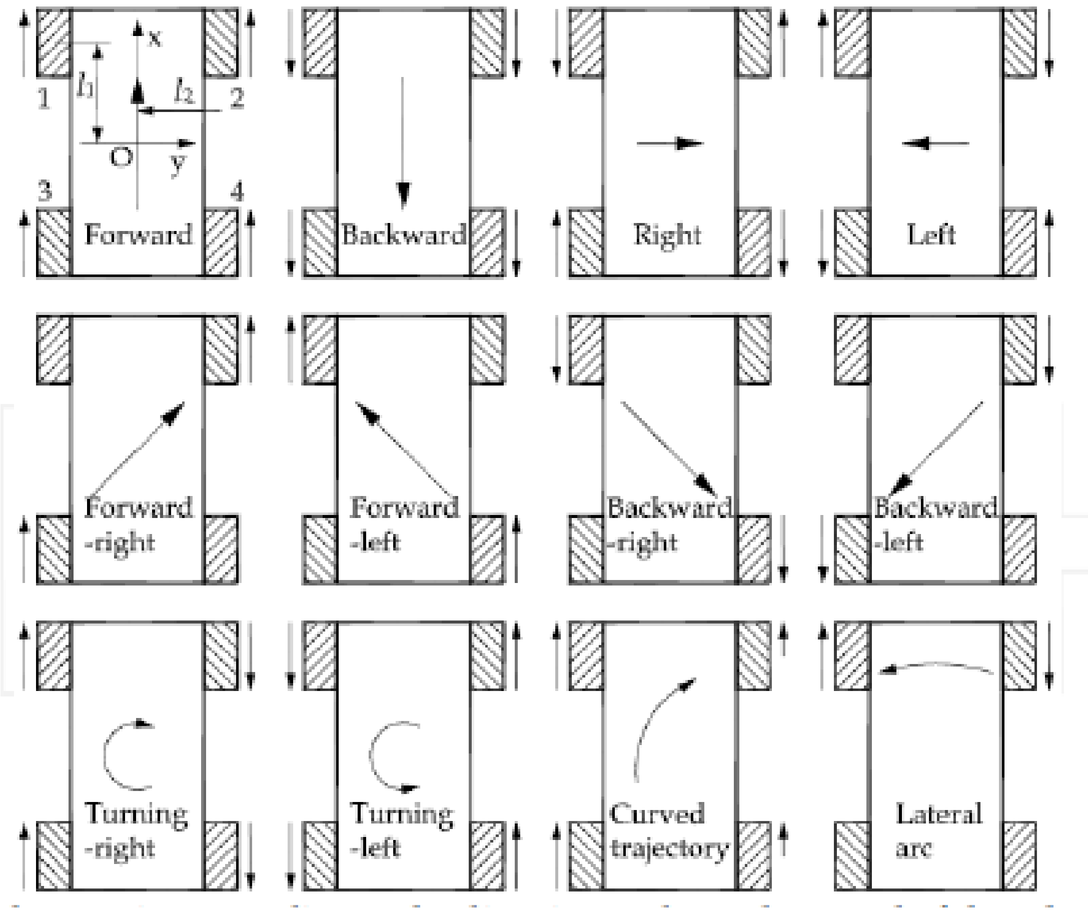


Figure 32. 12 kinds of basic movements of the robot (Doroftei et al., 2007, p. 522).

The relationship between velocity and angular velocity of the wheel is:

$$V_{iw} = \omega_{iw} \times R_{iw} \qquad (35)$$

, where V_{iw} (i=1, 2, 3, 4) is the velocity vector corresponding to wheel revolutions, R_{iw} is the radius of the wheel, $\omega_{iw}$is the angular velocity of the wheel. V_{ir} is the tangential velocity vector of the free roller touching the ground. It can be decomposed into two components which one is in X direction, the other is in Y direction (see the figure 33). (Wakchaure et al., 2011.)

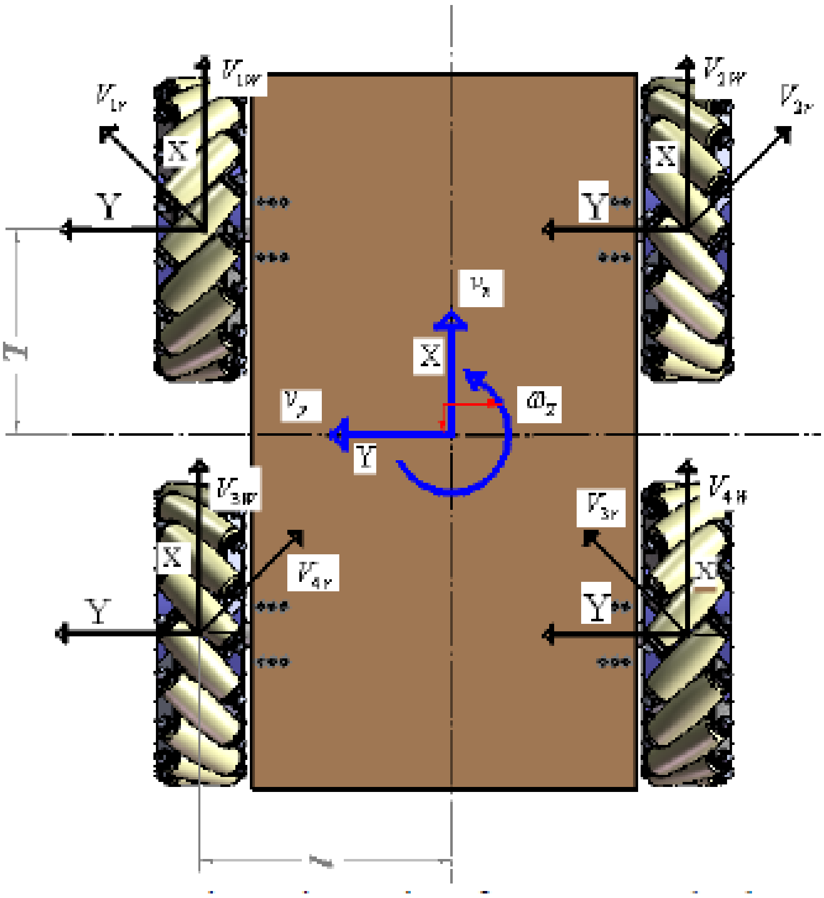


Figure 33. The analysis of force and velocity for the overall platform (Wakchaure et al., 2011).

Velocity analysis for wheel

$$V_{iX} = V_{iw} + V_{ir} cos45° \tag{36}$$

$$V_{iY} = V_{ir} sin45° \tag{37}$$

Velocity analysis for robot:

$$V_{iX} = v_X - l\omega_Z \tag{38}$$

$$V_{iY} = v_Y + L\omega_Z \tag{39}$$

, where v_{iX}, v_{iY} and ω_Z (i=1, 2, 3, 4) represent the velocity and angular velocity of the robot. L is the X axis distance from each wheel to the center of gravity, and l is the Y axis distance form each wheel to the center of gravity. (Wakchaure et al., 2011.)

Combining the velocity analysis for wheel and robot based on equation (36), (37), (38) and (39):

$$V_{1w} = v_X - v_Y - (L + l)\omega_Z$$
$$V_{2w} = v_X + v_Y + (L + l)\omega_Z$$
$$V_{3w} = v_X + v_Y - (L + l)\omega_Z \tag{40}$$
$$V_{4w} = v_X - v_Y + (L + l)\omega_Z$$

$$\begin{bmatrix} v_X \\ v_Y \\ \omega_Z \end{bmatrix} = \frac{1}{4} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l+L)} & \frac{1}{(l+L)} & -\frac{1}{(l+L)} & \frac{1}{(l+L)} \end{bmatrix} \cdot \begin{bmatrix} R_w\dot{\theta}_1 \\ R_w\dot{\theta}_2 \\ R_w\dot{\theta}_3 \\ R_w\dot{\theta}_4 \end{bmatrix} \tag{41}$$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \frac{1}{R} \cdot \begin{bmatrix} 1 & -1 & -(L+l) \\ 1 & 1 & (L+l) \\ 1 & 1 & -(L+l) \\ 1 & -1 & (L+l) \end{bmatrix} \cdot \begin{bmatrix} v_X \\ v_Y \\ \omega_Z \end{bmatrix} \tag{42}$$

In the global reference coordinates (X, Y, Z), the velocity of the robot would be:

$$v_X = \frac{R_w}{4}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \tag{43}$$

$$v_Y = \frac{R_w}{4}(-\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 - \dot{\theta}_4) \tag{44}$$

$$\omega_Z = \frac{R_w}{4(l+L)}(-\dot{\theta}_1 + \dot{\theta}_2 - \dot{\theta}_3 + \dot{\theta}_4) \tag{45}$$

In the stationery coordinate axis (x, y, z), the direction of the resultant motion would be:

$$\beta = \tan^{-1}(\frac{v_X}{v_Y}) \tag{46}$$

$$v_r = \sqrt{(v_X{}^2 + v_Y{}^2)} \tag{47}$$

, where β is the direction of motion and v_r is the resultant velocity of omni-directional platform with Mecanum wheels. (Wakchaure et al., 2011.)

The inverse kinematic equations can obtain the angular velocity of each wheel individually from the known motion of the robot. The forward kinematic equations are to display the velocity and rotation of robot through calculating the desired angular velocities of the four wheels. (Wakchaure et al., 2011.) According to equation (43), (44), and (45), the results shown as the following:

When $v_X \neq 0, v_Y = 0, \omega_Z = 0$, the robot will do the straight move in X direction. The angular velocity of each wheel is v_X/R_w.

When $v_X = 0, v_Y \neq 0, \omega_Z = 0$, the robot will do the straight move in Y direction. The angular velocity of wheel 1 and 4 is $-v_Y/R_w$. The angular velocity of wheel 2 and 3 is v_Y/R_w.

When $v_X = 0, v_Y = 0, \omega_Z \neq 0$, the robot will do the rotatory move via Os point. The angular velocity of wheel 1 and 4 is $-(l + L)\omega_Z/R_w$. The angular velocity of wheel 2 and 3 is $(l + L)\omega_Z/R_w$.

When $v_X = Vcos\varphi, v_Y = Vsin\varphi, \omega_Z = 0$, the robot will do the move with V velocity based on any angle φ which is built with Y axis. The angular velocity of wheel 1 and 4 is $\mathrm{V}(\mathrm{cos\varphi} - \mathrm{sin\varphi}/\mathrm{tan\alpha})/R_w$. The angular velocity of wheel 2 and 3 is $\mathrm{V}(\mathrm{cos\varphi} + \mathrm{sin\varphi}/\mathrm{tan\alpha})/R_w$.

2.3.2 Programing of the hardware in ROS

The ROS is a flexible framework for programming robot code. It can be used on Ubuntu Linux and worked on robotic platforms. This section is mainly for EPOS motor controller processing. There already are general “epos_hardware” packages on ROS GitHub to test one epos motor and then change some codes based on these packages to control the rest of motors together. (ROS, 2015.)

As mentioned before, firstly trying to control one part of EPOS motor controller. The EPOS controller is connected motor part with cables through USB directly to the computer (see the figure 34). It is meanwhile as being the master.

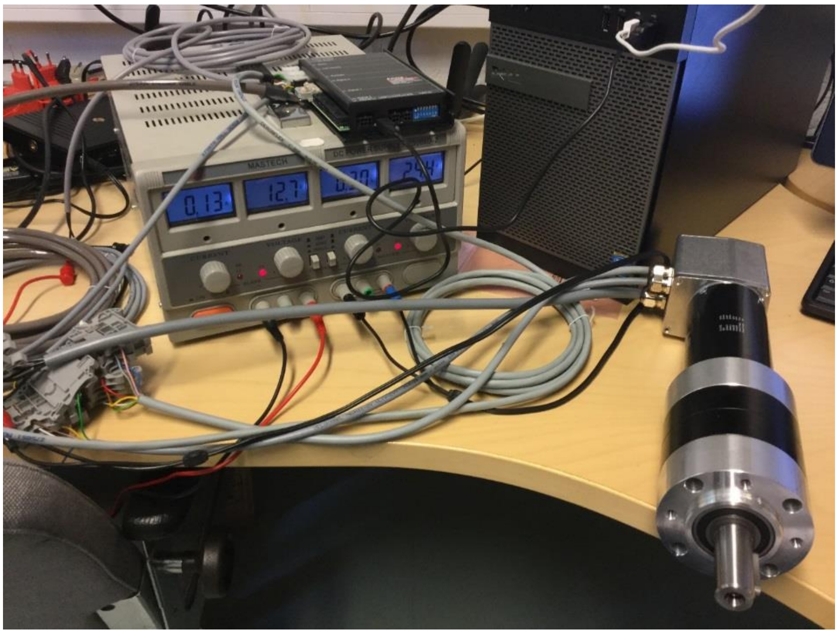


Figure 34. One set of motor controller.

CAN bus is a controller area network which supplies a protected communication channel to exchange up to 8 bytes between several network nodes. One set of motor part is done already, then the rest of motor controllers are connected in serial through CAN connections to a single USB port. The connection of four motor can be seen the figure 35.

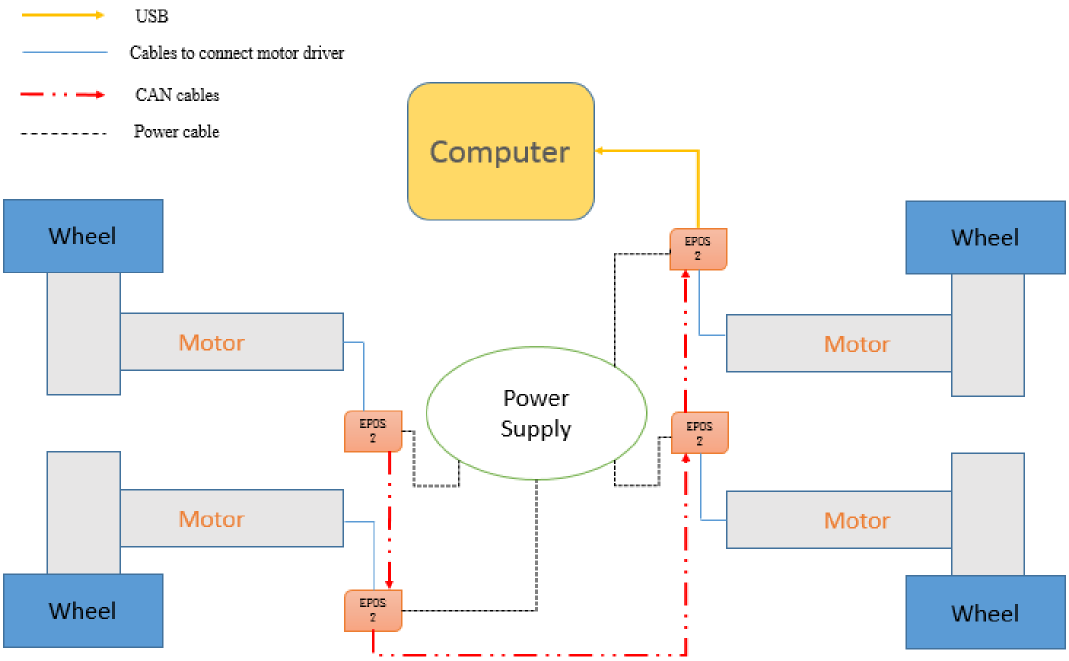


Figure 35. Four motor driver connection.

Each motor controller is used parallel power connection and presents a series of pins on the side. The first EPOS controller which is the master is set by pin number 1 and the following controllers by pin number 2, 3, 4 (see the figure 36). (Bianchi, 2013.)

Figure 36. 2 EPOS controllers by setting pin number 1 and 2.

After establishing the physical connections, the network must be set up via EPOS studio software. When connecting the master motor controller to the computer and then scanning for devices, it would be shown the all of EPOS controllers with node number which is set (see the figure 37). Finally, the controllers could be ready to program on Ubuntu Linux. (Bianchi, 2013.)

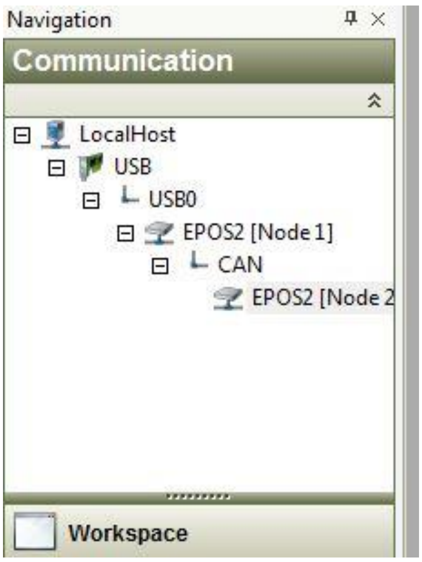


Figure 37. 2 EPOS2 controllers recognized on EPOS studio software.

The following figure 38 presents the general idea of how the motor controller works on Ubuntu Linux. There are two main packages in the workspace. 'epos_hardware_node' is to control four of the motors, both 'joy_node' and 'teleop_node' are simultaneously controlling the joystick. There is inverse kinematics equations programmed in the 'teleop_node' to contact with joystick. Joystick as subscriber is to give the command to each motor, while the each motor as publisher is going to show the speed and direction. Using the code 'list_devices.cpp', all the motors can be found and recognized on the terminal. Then combining the data of motors to the 'example.launch' which is launch file to launch the motor, and all the motors are ready to receive command. After that, publishing the command and give the speed to let each motor shaft rotate. The subscribe command is about joystick code which will be used in the future. Once the joystick code is testing and working well, it would be applied as input code. The wheels will be controlled by joystick and the velocity of each wheel will state.

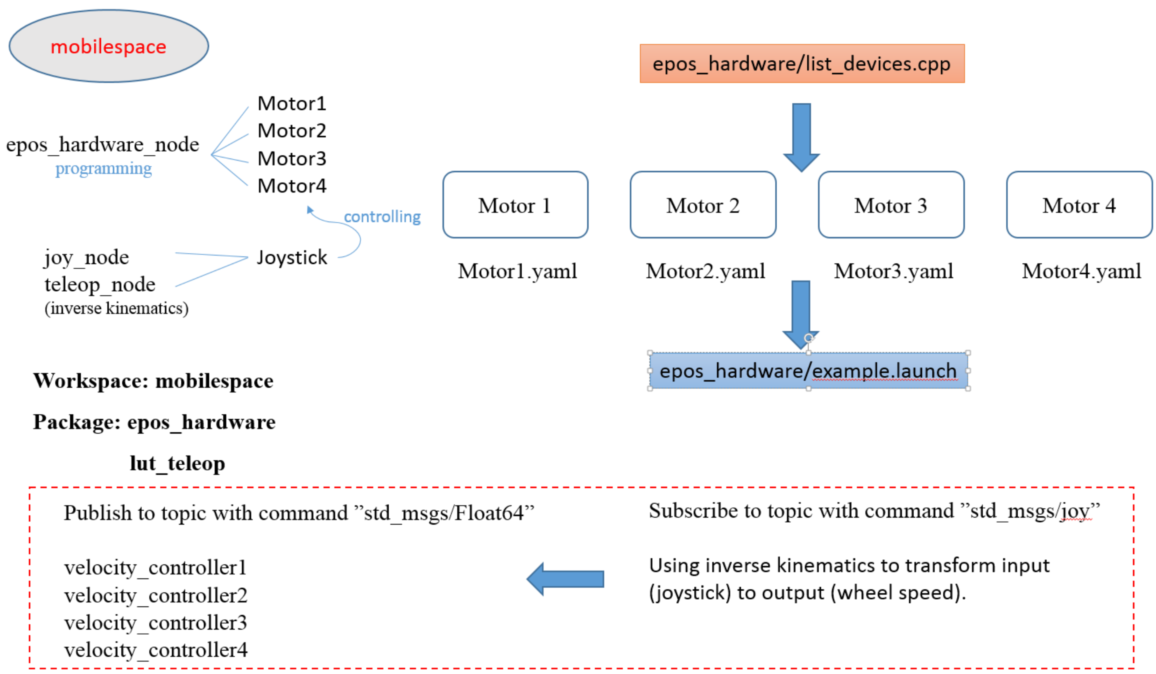
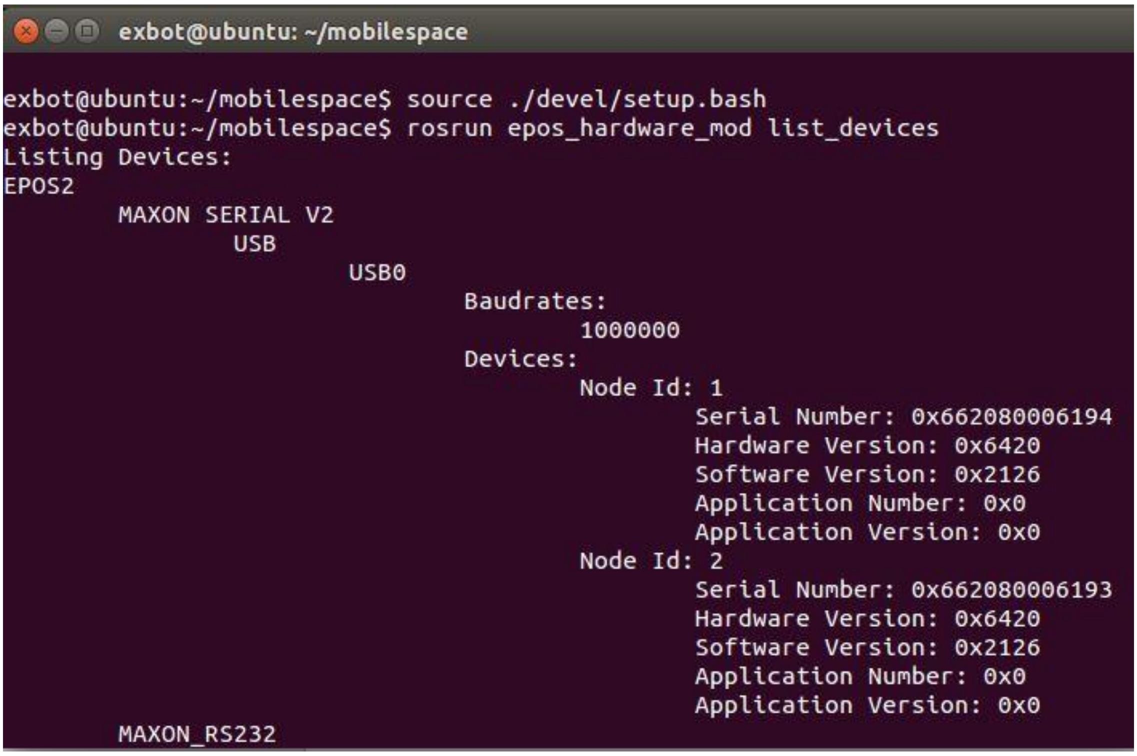


Figure 38. Code used on Ubuntu Linux.

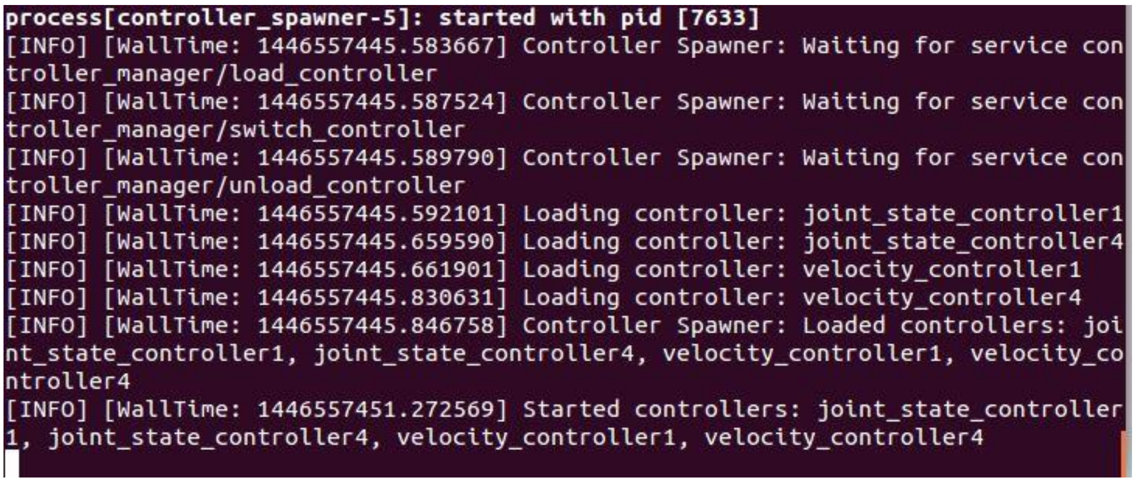
The following paragraph states two of the motor controllers setting and programming, and all of them would be the same steps. After recognizing two devices when using CANopen connection (see the figure 39). Input command 'roslaunch epos_hardware example.launch', the controllers are ready to rotate (see the figure 40). The launch file must be changed due to the number of devices used, the code would be shown later as appendix 2. The command <rostopic pub /velocity_controller/command “std_msgs/Float64” ‘10’> which means publish the given speed to the motor with value 10 (see the figure 41).



```
exbot@ubuntu: ~/mobilespace

exbot@ubuntu:~/mobilespace$ source ./devel/setup.bash
exbot@ubuntu:~/mobilespace$ rosrun epos_hardware_mod list_devices
Listing Devices:
EPOS2
        MAXON SERIAL V2
                USB
                        USB0
                                Baudrates:
                                        1000000
                                Devices:
                                        Node Id: 1
                                                Serial Number: 0x662080006194
                                                Hardware Version: 0x6420
                                                Software Version: 0x2126
                                                Application Number: 0x0
                                                Application Version: 0x0
                                        Node Id: 2
                                                Serial Number: 0x662080006193
                                                Hardware Version: 0x6420
                                                Software Version: 0x2126
                                                Application Number: 0x0
                                                Application Version: 0x0
        MAXON_RS232
```

Figure 39. Two devices recognized on ROS Ubuntu.



```
process[controller_spawner-5]: started with pid [7633]
[INFO] [WallTime: 1446557445.583667] Controller Spawner: Waiting for service con
troller_manager/load_controller
[INFO] [WallTime: 1446557445.587524] Controller Spawner: Waiting for service con
troller_manager/switch_controller
[INFO] [WallTime: 1446557445.589790] Controller Spawner: Waiting for service con
troller_manager/unload_controller
[INFO] [WallTime: 1446557445.592101] Loading controller: joint_state_controller1
[INFO] [WallTime: 1446557445.659590] Loading controller: joint_state_controller4
[INFO] [WallTime: 1446557445.661901] Loading controller: velocity_controller1
[INFO] [WallTime: 1446557445.830631] Loading controller: velocity_controller4
[INFO] [WallTime: 1446557445.846758] Controller Spawner: Loaded controllers: joi
nt_state_controller1, joint_state_controller4, velocity_controller1, velocity_co
ntroller4
[INFO] [WallTime: 1446557451.272569] Started controllers: joint_state_controller
1, joint_state_controller4, velocity_controller1, velocity_controller4
```

Figure 40. Two devices are ready to be set speed.

```
exbot@ubuntu: ~
exbot@ubuntu:~$ rostopic pub /velocity_controller1/command "std_msgs/Float64" '1
0'
publishing and latching message. Press ctrl-C to terminate
^C
exbot@ubuntu:~$
exbot@ubuntu:~$ rostopic pub /velocity_controller4/command "std_msgs/Float64" '1
0'
publishing and latching message. Press ctrl-C to terminate
```

Figure 41. Given speed command.

After all the motors rotate with the given speed successfully, it turns to be controlled by joystick with kinematics formula in the previous part (see the figure 42).

```
float width = 0.339075;          // half distance between left right wheels center unit in meter
float length = 0.582;          // half distance between front back wheels center
float wheelRadius = 0.127;      // radius of mecanum wheel


vel_wheel_1 = 1/wheelRadius*(vel_linear_x+vel_linear_y+(width+length)*vel_angular_z);

vel_wheel_2 = 1/wheelRadius*(vel_linear_x-vel_linear_y+(width+length)*vel_angular_z);

vel_wheel_3 = 1/wheelRadius*(vel_linear_x-vel_linear_y-(width+length)*vel_angular_z);

vel_wheel_4 = 1/wheelRadius*(vel_linear_x+vel_linear_y-(width+length)*vel_angular_z);
```

Figure 42. The velocity of four wheels presents as code.

At this moment, four motors can be controlled by joystick with constant speed respectively. Due to the big project, each section needs to be ready and contacted such as Mecanum wheels tested, manufactured shaft done. Thus, it would be taking a little bit longer time to get the expected idea.

3 DESIGN OF SUSPENSION SYSTEM

Suspension is applied to connect the wheels to the vehicle to ensure the bounce after spring compression and extension. It seems to make a robot as simple as possible, so a suspension system is rarely needed in most cases. However, in this case, different working place and four wheels configuration are considered, a small suspension system was utilized to avoid damage things like joints and gears, and the effect of sensor data if the terrain is uneven.

The suspension system provides a smooth ride and prevents the robot body and frame from ground shocks. It can support the weight as well so that giving the stability of the robot. Dependent and independent suspension are the two geometries mainly applied. Dependent suspension can be called conventional suspension system which means there is the rigid axle on each side mounted on two wheels. When one wheel comes upon a bump, a slight change of the other wheel is caused by its upward movement. Independent suspension means each wheel is utilized. When one wheel sinks, there is not much influence on the other wheel. (Lajqi et al., 2013, p. 102-103.)

There are such kinds of springs can be selected, but a shock absorber is finally decided to use because it restricts spring compression and extension movements to flatten the path of the robot. It is bought from market (see the figure 43). It is one of the main part suspension system for motorcycle, but we do some changes about size and use shaft to connect with wheel. It is just an idea, and the suspension part will be simulated and finalized based on the whole design.

Figure 43. Shock absorber (adjustable) with 10 mm holes on each side and the length is about 265 mm (Shock absorber, 2015).

3.1 Structure design 1

For the first time, the suspension system was designed as simple as possible without considering the whole situation (see the figure 44). The Mecanum wheel and gearhead were connected with shaft, and the shock absorber was fixed to the shaft. However, this idea made the shock absorber might not carry the whole weight and also axial force when the wheel is moving.

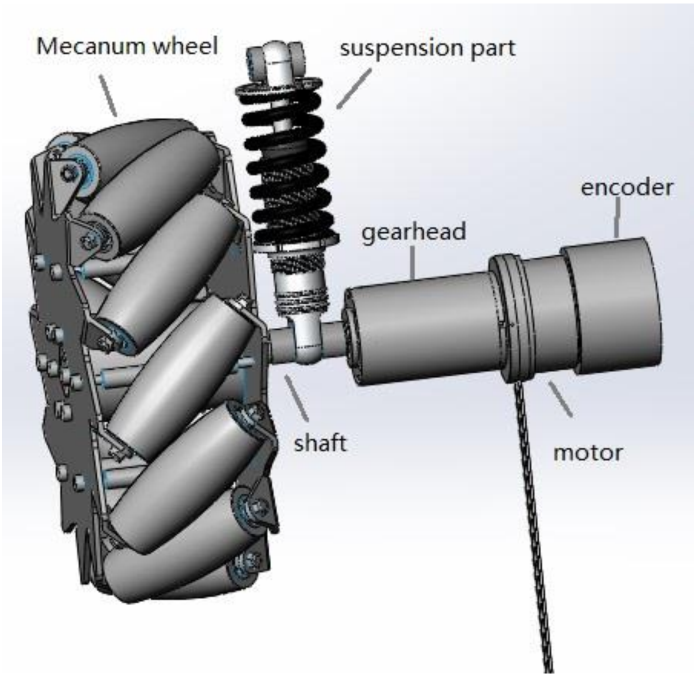


Figure 44. Design 1.

3.2 Structure design 2

Due to the reason as mentioned before, the suspension part was redesigned and added one leg which has 30 degree to the vertical line to support the whole chassis as the same job as shock absorber (see the figure 45). However, when analyzing the force, this kind of structure could support the chassis, but it had better to need more support part in order to make the whole system stable and safe.

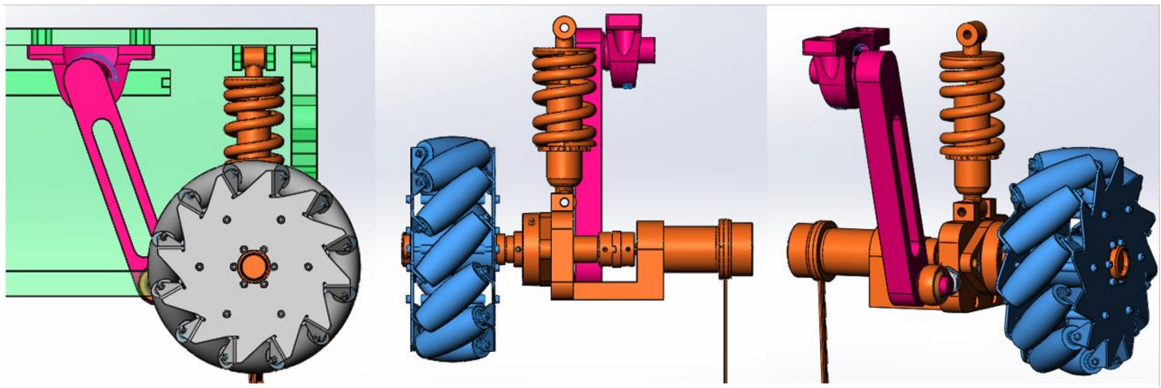


Figure 45. Design 2.

3.3 Structure design 3

In the meantime, a system as similar as car structure was designed (see the figure 47). Surely, it was not a best choice for this case, because it would make the whole system more complicate.

The typical suspension system of a vehicle (see the figure 46) has several basic parts needed to consider. The main part of the suspension is the steering knuckle which acts a support of spindle or bearing to connect with wheel hub, bearings and wheel assembly. The control arms have upper one and lower one are movable, but they can fasten the steering knuckle to the frame part. The radius rod is a connection between the steering knuckle and the frame as well. The steering gear which is also called tie rod, connects the steering knuckle to the rest of the steering system. The ball joints that are located upper and lower control arm joint separately can provide the motion of control arm and steering knuckle with up and down, left and right. Coil spring provides a weight of the vehicle support and allows the motion of control arm and wheel. And the shock absorber shows a performance about maintaining the suspension from continuing to bounce during the process of spring compression and extension. (Suspension System Fundamentals, 2006, p. 1284-1289.)

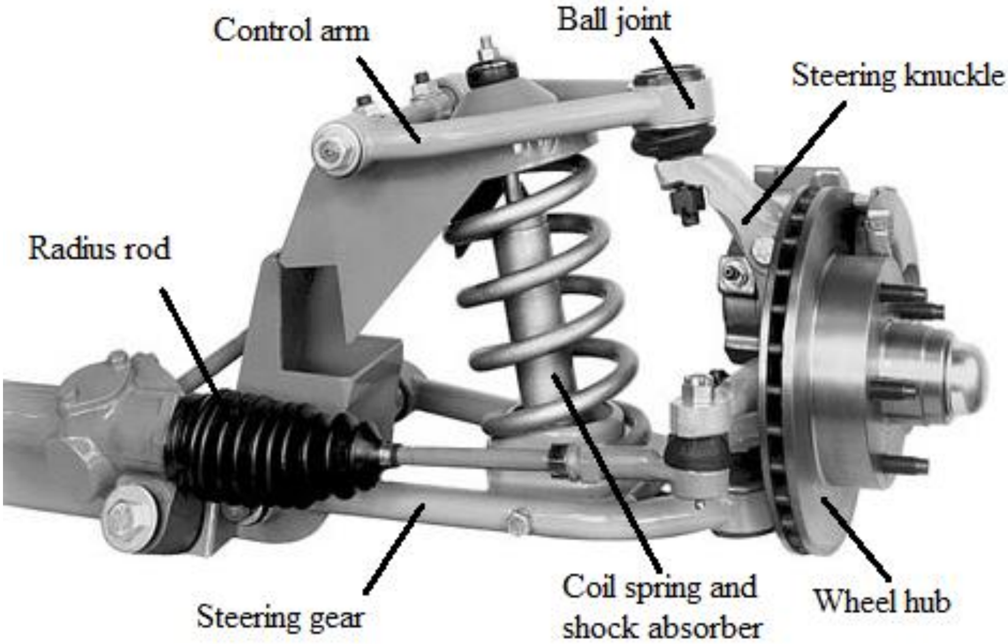


Figure 46. Mustang independent front suspension system (Suspension Parts, 2002).

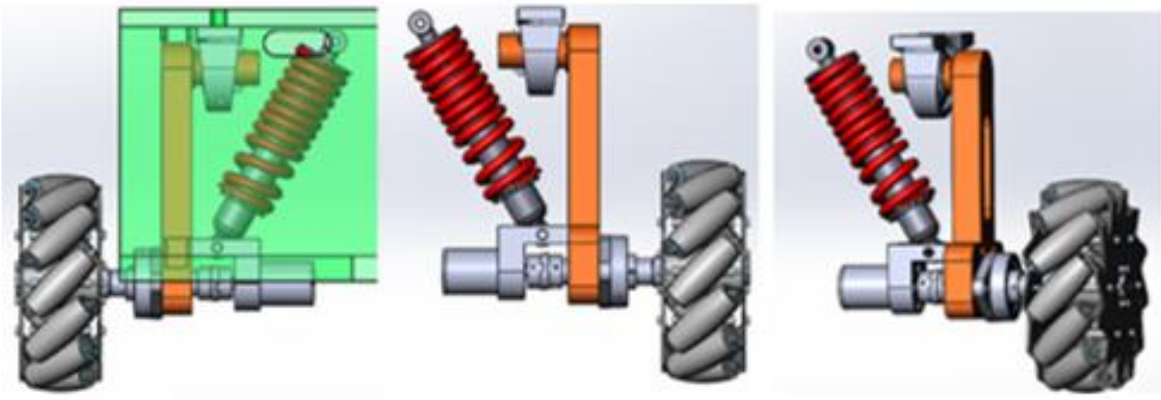


Figure 47. Design 3.

3.4 Structure design 4

Thus, one more leg was added in order to make the system more stable like the following image shown (see the figure 48). Figure 49 is for the whole general robot chassis and layout of the components. There are four holes in the chassis, two holes from the bottom are to let the wires into the interior of the chassis to connect the motor controllers and battery. It has still problem that this structure make the whole chassis higher. The height from the ground to the top of chassis as calculated was about half meter.

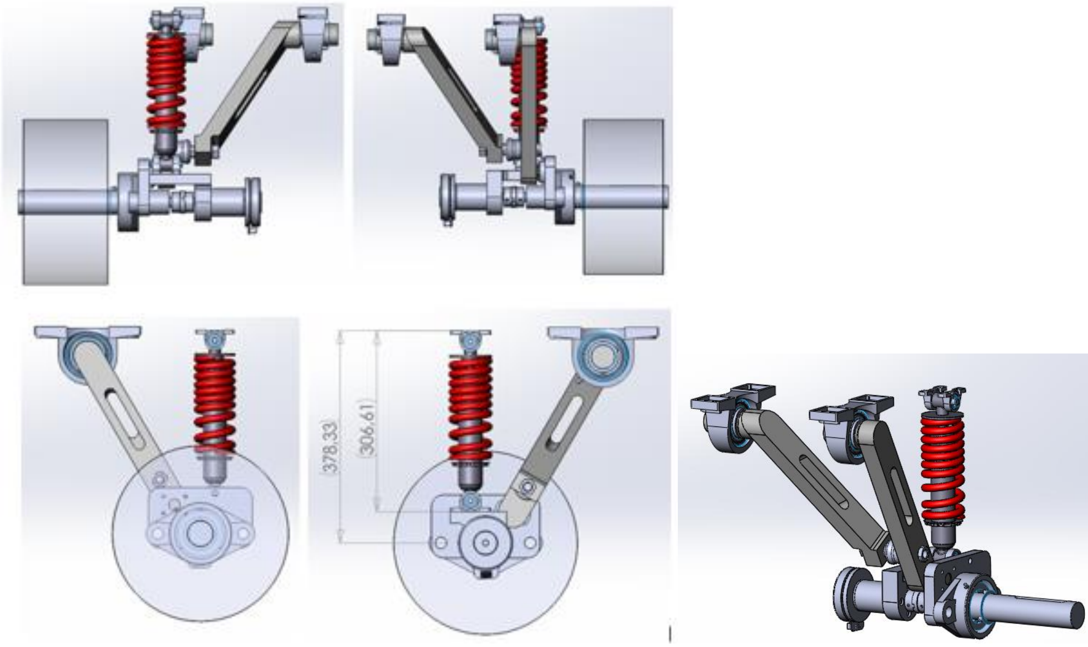


Figure 48. Design 4.

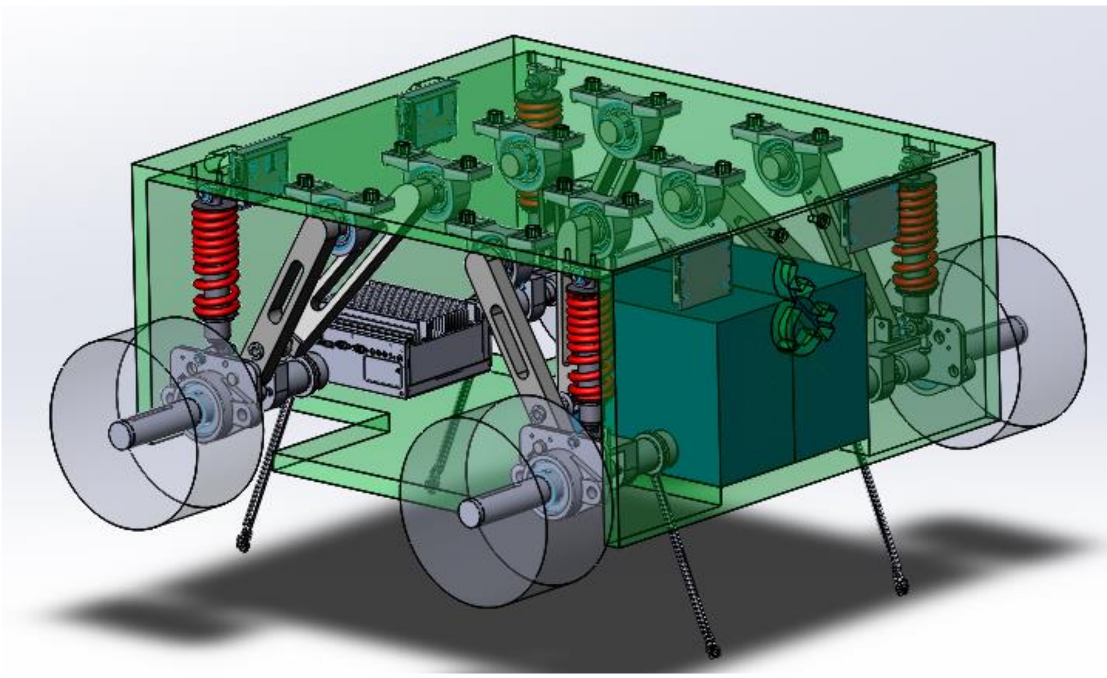


Figure 49. General robot chassis with required components.

3.5 Structure design 5

Due to height problem, the shock absorber was made much lower and finally got the height around 370 mm (see the figure 50). The whole suspension system with other components can be seen in the figure 51. The ball bearings were mounted to the chassis and the small ones were fixed to both support part and chassis in order to make the limbs and the shock absorber have degree of freedom. The shaft collar was exploited to lock the shaft while the wheel rotates.

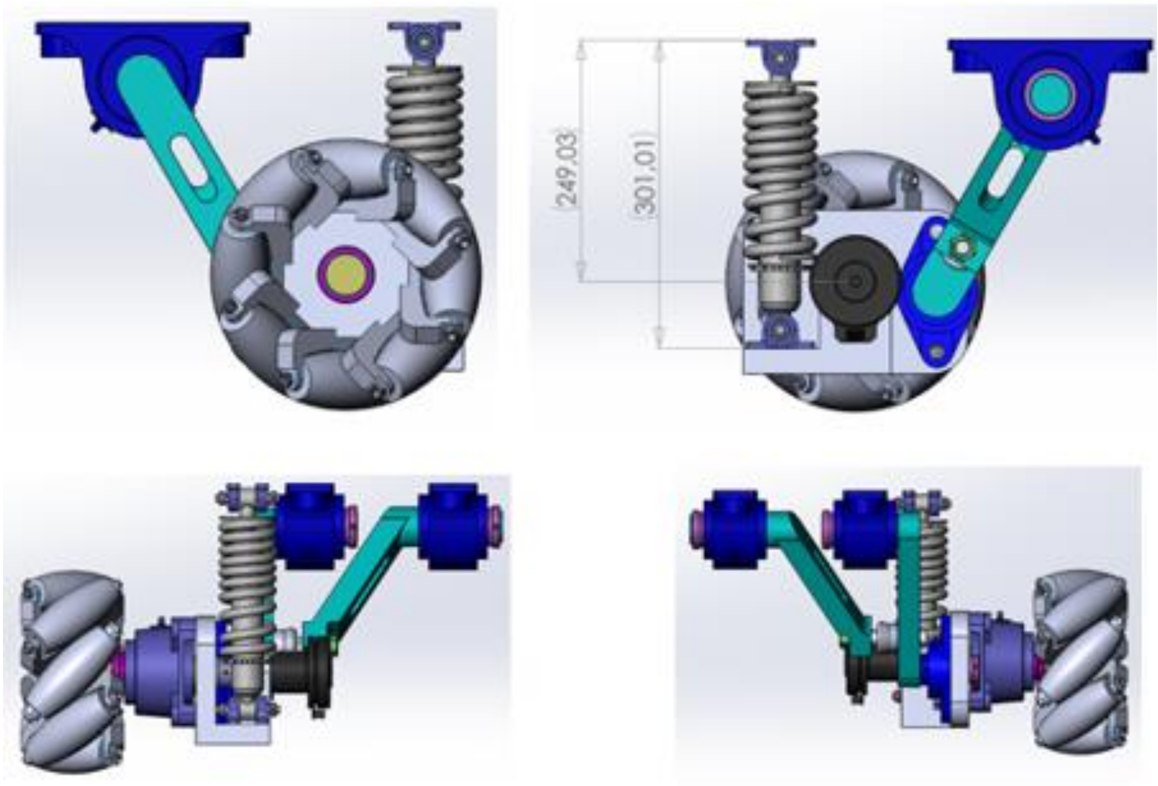


Figure 50. The arrangement of Design 5.

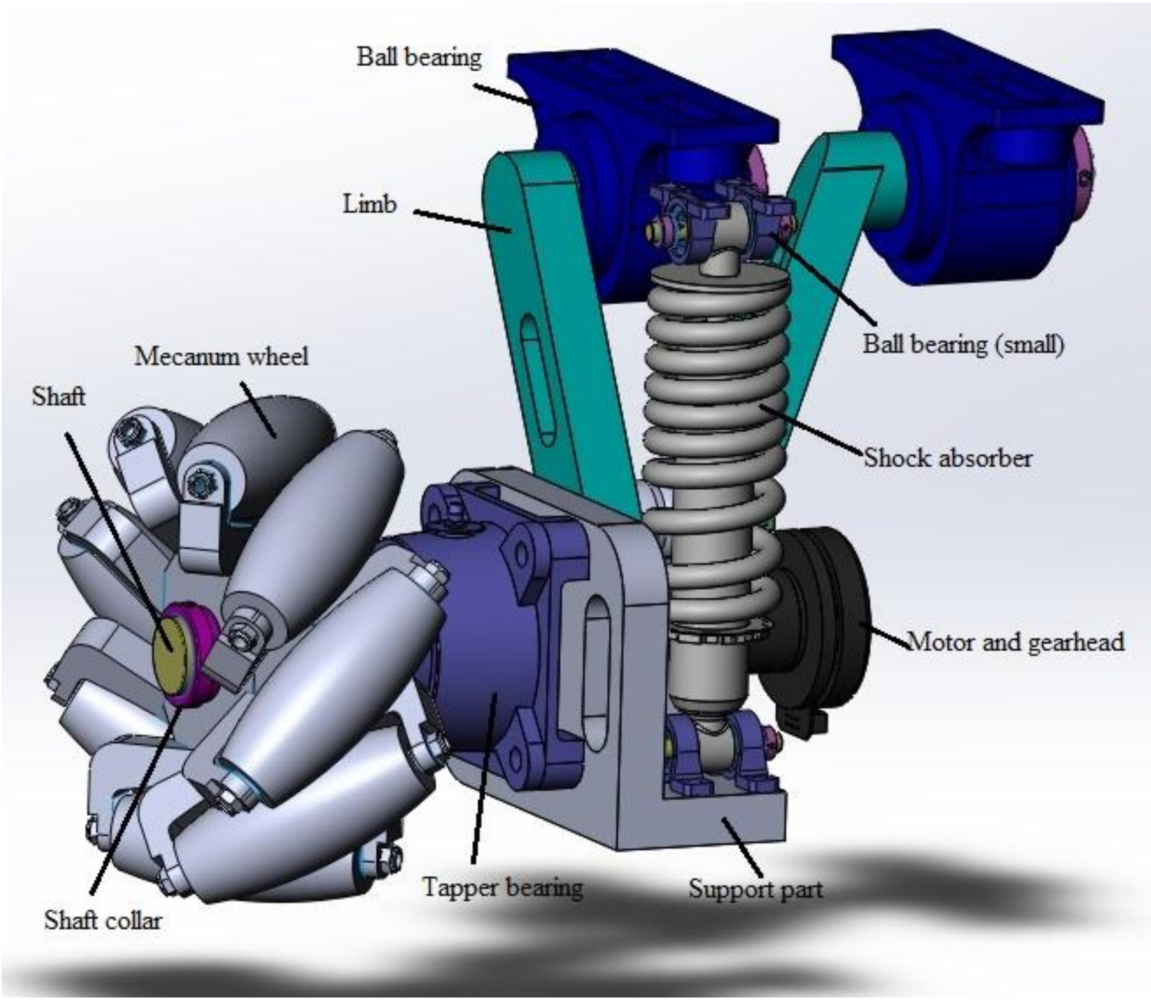


Figure 51. Final design of suspension system.

3.6 Results of suspension system

Although the height of the chassis is already reduced as much as possible, it still does not get what we expected before. After discussion, the idea of putting suspension system on the robot chassis is given up. To make the chassis structure easier, the design of transmission system without the suspension system is selected.

4 CONCLUSIONS

In this report, Mecanum wheel and its configuration introduce firstly in order to get more theoretical knowledge. The selection of wheel makes the project become interesting and challenge. In the later chapter, the inverse kinematics analysis of Mecanum wheels have been solved in real time, and the pose control of the mobile platform has been realized in the Cartesian space. The design of traction system presents the whole platform with motors, wheels, gearboxes and other components. Suspension system is planned to utilize at the beginning of the project idea, but it is cancelled at last. The reason is that the size of platform did not meet the expected idea. It is sad that we spent a lot of time to design the suspension system, but the idea of the design would be helpful in the future. Furthermore, there is a part about motor control which is the main section in the control system of mobile robot.

SolidWorks is an important software to be used in this research, because it would be a good way to guarantee the accuracy of each transmission components assembly. ROS is a necessary tool to program the mobile robot, not only in the motor control, but also in the overall control system.

This project has been taking almost one year, as mentioned before, it combine about six sections. With the cooperation of our team, the model of mobile robot is done via SolidWorks (see the figure 52). Currently, the products about traction system from market are received and the shafts and structures of the chassis will manufactured soon.

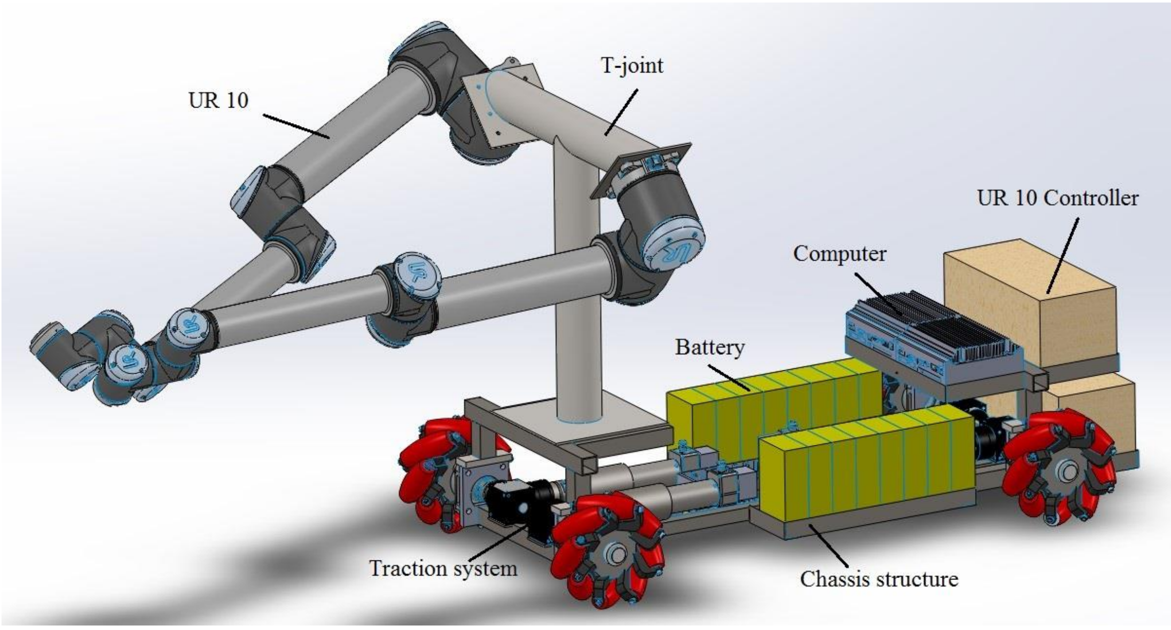


Figure 52. Model of mobile robot with main parts.

In the future work, the assembly plan will start after getting all the products. The motors can be controlled with constant value successful so far, the accurate pose control of the entire robot with the torque control in the prescribed maneuverable direction will be studied. For the purpose of moving the mobile robot smoothly with variable speed of each wheel based on inverse kinematics analysis.

REFERENCES

Bianchi, F. 2013. CAN serial connection of Maxon Motors in LabVIEW, through a single USB port. [Referred 01.09.2015]. Available in PDF-file: http://thrish.org/wp/wp-content/uploads/2013/06/CAN-Tutorial.pdf.

Bräunl, T. 2006. Embedded Robotics. Mobile Robot Design and Applications with Embedded Systems. Second Edition. Springer Berlin Heidelberg New York, held at Germany, 2006. Pp. 96-148.

Doroftei, I., Grosu, V. & Spinu,V. 2007. Omnidirectional Mobile Robot-Design and Implementation. Maki, K. H. Bioinspiration and Robotics Walking and Climbing Robots. I-Tch Education and Publishing, 01 September, 2007. Pp. 511-528.

Doroftei, I. & Stirbu, B. 2010. Design, Modeling and Control of an Omni-directional Mobile Robot. Solid State Phenomena. Vols. 166-167. Pp. 173-178.

Dudek, G. & Jenkin, M. 2000. Mobile Robot Hardware. Computational Principles of Mobile Robotics. The Press Syndicate of the University of Cambridge, held at Cambridge, United Kingdom February, 2000. Pp. 13-15.

EPOS. 2015. [Maxon Motor webpage]. Updated 2015. [Referred 05.06.2015]. Available: http://www.maxonmotor.com/maxon/view/content/EPOS-Detailsite?etcc_med=Product&etcc_cmp=%2fmaxon%2fview%2fcontent%2fEPOS-Detailsite%3fnoMobile&etcc_cu=onsite&etcc_var=[com]%23en%23_d_&etcc_plc=product_index&noMobile.

Goris, K. 2004. Autonomous Mobile Robot Mechanical Design [web document]. Brussel: 2004-2005 [Referred 05.02.2015]. Academic degree thesis. Vrije Universiteit Brussel, Faculty of Engineering, Department of Mechanical Engineering. 88 p. Available in PDF-file: http://mech.vub.ac.be/multibody/final_works/ThesisKristofGoris.pdf.

Introduction to Robots and Robotics [web document]. Published 2010 [Referred 15.10.2015]. Available:
http://www.robotplatform.com/knowledge/Introduction/Introduction_to_Robots.html.

Lajqi, S., Pehan, S., Lajqi, N., Gjelaj, A., Psenicnik, J. & Emin, S. 2013. Design of Independent Suspension Mechanism for a Terrain Vehicle with Four Wheels Drive and Four Wheels Steering. Annals of Faculty Engineering Hunedoara-International Journal of Engineering, held at Romania, 2013. Pp. 101-108.

Luo, H. 2015. Information about Mecanum Wheels to my thesis [company e-mail]. Receivers: Juha Koivisto, Ming Li (cc). Sent 04.05.2015, 13.24 o'clock (GMT +0200). Attachment:"Mecanum Wheel. pdf".

Maxon EC motor. 2015. [Maxon Motor webpage]. Updated 2015. [Referred 05.06.2015]. Available: http://www.maxonmotor.com/medias/sys_master/root/8811457609758/375711-Getting-Started-En.pdf.

Maxon Motor product. 2015. [Maxon Motor webpage]. Updated 2015. [Referred 05.06.2015]. Available: http://www.maxonmotor.com/maxon/view/catalog.

Neal, A.J. 2010. Tips for Selecting DC Motors for your Mobile Robot [web document]. Servo Magazine: January, 2010 [Referred 13.04.2015]. Pp. 33-37. Available in PDF-file: http://www.servomagazine.com/uploads/issue_downloads/pdf/Tips%20For%20Selecting%20DC%20Motors%20For%20Your%20Mobile%20Robot.pdf.

Piccirillo, J. 2009. The Art and Science of Selecting Robot Motors [web document]. Huntsville: February 2009 [Referred 13.04.2015]. Academic degree thesis. University of Alabama in Huntsville, Department of Electrical and Computer Engineering. 42 p. Available in PDF-file: http://www.budgetrobotics.com/resources/Selecting_Robot_Motors_v21.pdf.

PowerGear. 2015. [Graessner webpage]. Updated April 2015. [Referred 07.06.2015]. Available: http://www.graessner.de/en/products/powergear.html.

ROS. 2015. [ROS webpage]. [Referred 20.06.2015]. Available: http://www.ros.org.

Rotex coupler. 2015. Torsionally flexible coupling with T-PUR. [KTR webpage]. [Referred 07.06.2015]. Available: https://www.ktr.com/fileadmin/ktr/media/Tools_Downloads/kataloge/ktr-kat99ep02.pdf.

Shock absorber. 2015. [www- data sheet]. (Publishing place unknown): Tuontitukku. [Referred 10.08.2015]. Available: http://tuontitukku.fi.

Suspension Parts. 2002. [www- data sheet]. (Publishing place unknown): Performance Online. [Referred 10.08.2015]. Available: https://www.performanceonline.com/1947-59-CHEVY-GMC-3100-I.F.S.-SUSPENSION-SYSTEM-MUSTANG-II-TYPE.

Suspension System Fundamentals. 2006. Modern Automotive Technology-Fundamentals, service, diagnostics. Pp. 1280-1301.

Tapered Roller Bearings. 2015. [Timken webpage]. [Referred 07.06.2015]. Available: http://www.timken.com/en-us/products/bearings/productlist/roller/tapered/Pages/default.aspx.

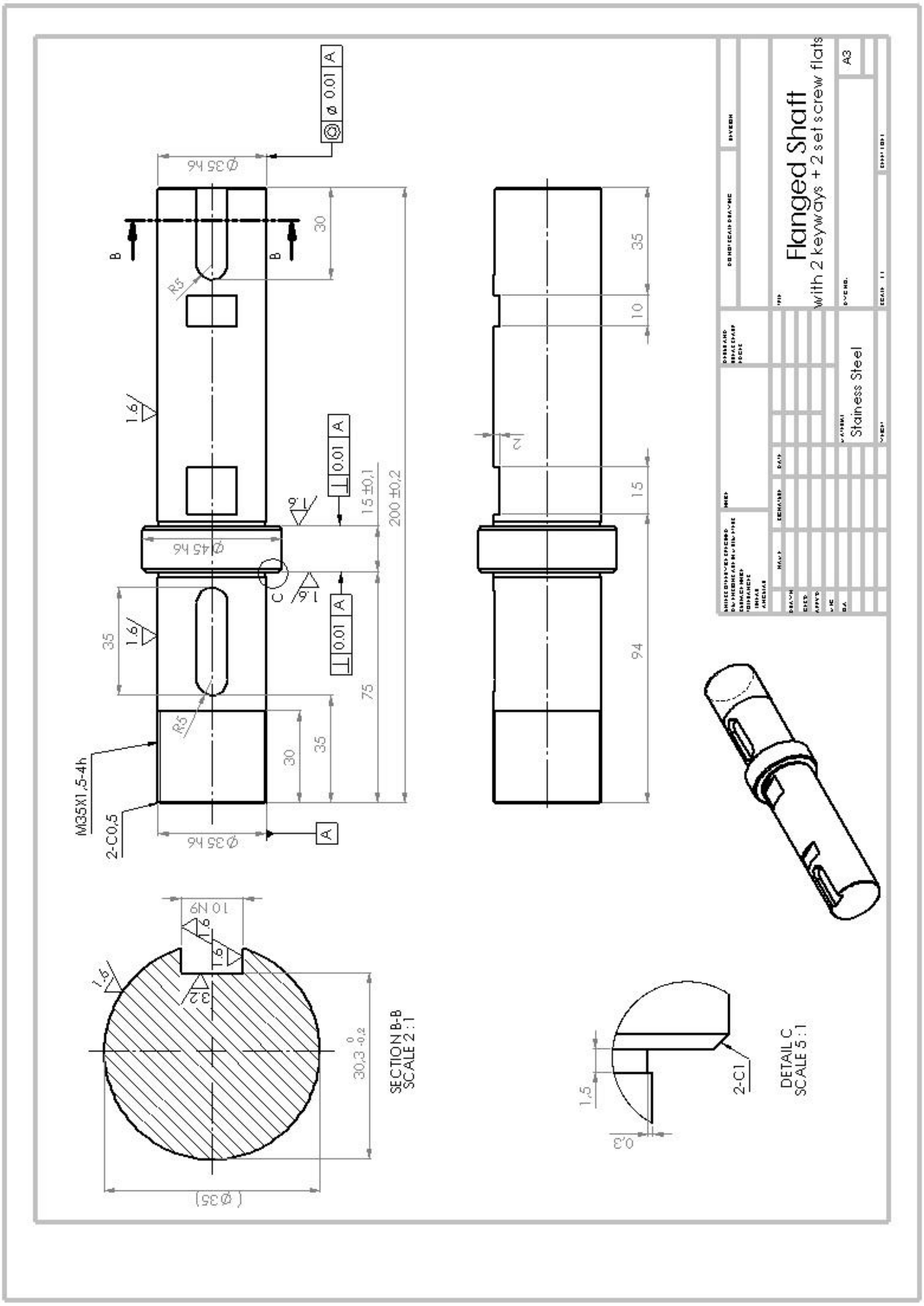
Twin FU Nut. 2008. [Fuji Selmltsu webpage]. Updated 2015. [Referred 07.06.2015]. Available: http://www.fun.co.jp/english/products/twin01.html.

Wakchaure, K. N., Bhaskar,S.V., Thakur, A.G. & Modak, G.S. 2011. Kinematics Modelling of Mechanum Wheeled Mobile Platform.

Winston Battery. 1994. [www- data sheet]. (Publishing place unknown): EV-power. [Referred 10.06.2015]. Available: http://www.ev-power.eu/Winston-40Ah-200Ah/WB-LYP60AHA-LiFeYPO4-3-2V-60Ah.html.

254mm Mecanum Wheels with PU roller. 2015. [Nexusrobot webpage]. [Referred 30.03.2015]. Available: http://www.nexusrobot.com/product.php?id_product=188.

Manufactured shaft of the appendix 1.



Launch file with four devices of the appendix 2.

```
<launch>
  <!--node name="speed_pub" pkg="epos_hardware_mod" type="speed_pub" /-->
  <param name="robot_description" textfile="$(find
epos_hardware_mod)/launch/example.urdf" />
  <node name="epos_hardware" pkg="epos_hardware_mod" type="epos_hardware_node"
args="my_joint_actuator1 my_joint_actuator2 my_joint_actuator3 my_joint_actuator4">
    <rosparam command="load" file="$(find epos_hardware_mod)/launch/motor1.yaml" />
    <rosparam command="load" file="$(find epos_hardware_mod)/launch/motor2.yaml" />
    <rosparam command="load" file="$(find epos_hardware_mod)/launch/motor3.yaml" />
    <rosparam command="load" file="$(find epos_hardware_mod)/launch/motor4.yaml" />
  </node>
  <node name="lut_teleop" pkg="lut_teleop" type="lut_teleop_node"/>
  <node name="joy" pkg="joy" type="joy_node"/>
  <node name="controller_spawner" pkg="controller_manager" type="spawner"
respawn="false" output="screen" args="joint_state_controller1 joint_state_controller2
joint_state_controller3 joint_state_controller4 velocity_controller1 velocity_controller2
velocity_controller3 velocity_controller4"/>
  <param name="velocity_controller1/type"
value="velocity_controllers/JointVelocityController" />
  <param name="velocity_controller1/joint" value="test_joint1" />
  <param name="velocity_controller2/type"
value="velocity_controllers/JointVelocityController" />
  <param name="velocity_controller2/joint" value="test_joint2" />
  <param name="velocity_controller3/type"
value="velocity_controllers/JointVelocityController" />
<param name="velocity_controller3/joint" value="test_joint3" />
```

```
  <param name="velocity_controller4/type"
value="velocity_controllers/JointVelocityController" />
  <param name="velocity_controller4/joint" value="test_joint4" />
  <!--param name="position_controller/type"
value="position_controllers/JointPositionController" />
<param name="position_controller/joint" value="test_joint1" /-->
<!--param name="position_controller/type"
    value="position_controllers/JointPositionController" />
  <param name="position_controller/joint" value="test_joint2" /-->
  <!--param name="position_controller/type"
value="position_controllers/JointPositionController" />
  <param name="position_controller/joint" value="test_joint3" /-->
  <!--param name="position_controller/type"
value="position_controllers/JointPositionController" />
  <param name="position_controller/joint" value="test_joint4" /-->
    <param name="joint_state_controller1/type"
value="joint_state_controller/JointStateController" />
  <param name="joint_state_controller1/publish_rate" value="1000" />
  <param name="joint_state_controller2/type"
value="joint_state_controller/JointStateController" />
  <param name="joint_state_controller2/publish_rate" value="1000" />
  <param name="joint_state_controller3/type"
value="joint_state_controller/JointStateController" />
  <param name="joint_state_controller3/publish_rate" value="1000" />
  <param name="joint_state_controller4/type"
value="joint_state_controller/JointStateController" />
  <param name="joint_state_controller4/publish_rate" value="1000" />
</launch>
```