



Budapest University of Technology and Economics
Faculty of Mechanical Engineering
Department of Applied Mechanics

Control and Parameter Estimation Problems of Autonomous Transport Robots

Final Project

Made by:
Bálint CSATÓ
Supervisor:
Gergely GYEBRÓSZKI

Budapest, 2018

Contents

Abstract	2
1 Introduction to Autonomous Transport Robots	3
1.1 History of transportation	3
1.2 Autonomous vehicles	3
1.3 Application of autonomous transport robots	6
2 Mobile Robots	9
2.1 Classification of mobile robots [2][1]	9
2.1.1 Based on the medium in which the robot moves	9
2.1.2 Based on the locomotion mechanism	9
2.2 Legged Mobile Robots	10
2.3 Wheeled Mobile Robots	11
2.3.1 Wheel design	12
2.3.2 Wheel configurations	12
3 The Mobile Robot	16
3.1 Selecting a mobile robot configuration	16
3.2 The Hardware	17
3.2.1 Omni wheel	17
3.2.2 Gear Motor	17
3.2.3 Case	18
3.2.4 Electronics	19
3.2.5 Schematic diagram of the electronics	21
3.3 DC motor speed analysis	22
3.3.1 System modelling	22
3.3.2 Step response analysis of the motors	23
3.3.3 Controller design using blackbox model [18]	25
4 Motion Analysis of The Robot	31
4.1 Kinematics of three-wheeled omnidirectional robot	31
4.1.1 Basic relationships	31
4.1.2 2D Simulations	33
4.2 Dynamic analysis of DC motors	37
4.2.1 DC motor parameter identification	37
4.2.2 Validation of the parameter estimation	41
4.2.3 Rolling resistance measurements	41
4.3 Mechanical model of the robot	42
4.3.1 Model of the robot case	42
4.3.2 Simulations of motor states	46
4.3.3 Comparision with measurements	49
4.3.4 Mass estimation	51
Summary	53
References	54

Abstract

Autonomous vehicle technology offers the possibility of fundamentally changing transportation. Equipping cars and light vehicles with this technology will likely reduce crashes, energy consumption, and pollution and reduce the costs of congestion, as well.....
blabalblablablalba lablblablablalbalb labllablblalbalblalbal

1 Introduction to Autonomous Transport Robots

1.1 History of transportation

Since the early days of the human civilization, one of the most common problem has been the transportation. The first major development was the domestication of animals what made it possible to transport more and heavier loads or humans themselves in order to achieve greater speed and duration. The next substantial invention was the wheel which increased the efficiency of animal transport introducing the concept of vehicles. Until the Industrial Revolution, the water transport facilities proved to be the most effective way. With the development of the combustion engines and automobiles around the end of the 19th century, road transport became competitive again. Today trucks transport cargos over thousands of miles, but light items are still being carried in wheelbarrows or trailers attached to more compact vehicles.



Figure 1.1: Evolution of Transportation

One of the most dominant development tendency in almost every field of the industry is the automation of processes and procedures. Automation means the reduction of human intervention in the operation of machines. The benefit of automation include labor savings, savings in electricity costs, savings in material costs, and improvements to quality, accuracy and precision. The most trending application of the automation regarding the transportation is the autonomous cars on the roads.

1.2 Autonomous vehicles

An autonomous car which is also known as a self-driving car or a driverless car, is a vehicle that can operate without a certain amount of human control using its sensors to monitor the informations coming from the environment. Great variety of techniques are used to detect the surroundings, such as radar, laser light, GPS, odometry and computer vision. The collected information provides a decent input to the control system which is the soul of the self-driving cars. Several control problems have to be solved continuously during driving on the roads, such as identifying obstacles and avoiding them, following transportation rules or navigating along a predefined path.

Autonomous vehicle technology offers the possibility of fundamentally changing transportation. Equipping cars and light vehicles with this technology will likely reduce traffic collisions, thus the resulting injuries and the related costs including less need for insurance. Autonomous cars are predicted to increase the traffic flow and the mobility of citizens. Reduced traffic congestion and the improvements in traffic flow due to widespread use of autonomous cars will also result in better fuel efficiency and reduced pollution in cities.

In spite of the various benefits also several issues exist such as technology challenges, disputes concerning liability, customer concerns about the safety of driverless cars, risk of loss of privacy and security concerns against hackers, risk of negative effects on the society and economy and finally moral issues when car's software is forced to choose between multiple harmful alternatives during an inevitable accident. [4]

The autonomous driving technology can be most easily conceptualized using a five-part continuum suggested by the National Highway Traffic Safety Administration (NHTSA), with different benefits of the technology realized at different levels of automation:

- **Level 0:** The human driver is in complete control of all functions of the car.
- **Level 1:** One function is automated.
- **Level 2:** More than one function is automated at the same time (e.g., steering and acceleration), but the driver must remain constantly attentive.
- **Level 3:** The driving functions are sufficiently automated that the driver can safely engage in other activities.
- **Level 4:** The car can drive itself without a human driver.[3]

From engineering point of view, the most important issues are the technological challenges, i.e. how to design a driverless vehicle system which is capable of handling vehicle's performance like human in all possible conditions. An autonomous vehicle is a combination of sensors and actuators, sophisticated algorithms executed as a software on a processor. The sensory system can be classified into three different classes:

- **Navigation and guidance:** the system which determines – where you are, where you want to go, and how do you get there. In the most common applications different techniques such as GPS, compass and dead reckoning are used.
- **Driving and Safety:** Directing the vehicle and making sure that vehicle follows the rules of the road. The autonomous car must be able to see and interpret what is in front of it when it is going forward (and behind when in reverse). It is also necessary to see what is on either side, in other words, it needs a 360° view. A set of video cameras is an obvious choice by which location of lanes and surrounding objects or markers on the road can be determined.
- **Performance:** managing car's internal system. Several application specific, unique circuit boards and subsystems are added to a conventional vehicle to provide the functions needed for autonomous operation. Much of the system-level operation involves measuring and managing the power requirements to control power, overall consumption, and thermal dissipation.

The concept of driverless cars made a lot of attention over the past ten years. Generally speaking it can be said that one of the most determining development directions in the automotive industry are related to automated solutions. Many people still do not believe that there could be such a possibility with fully driverless vehicles due to the fact that there are so many parameters in the driving function that have to be controlled simultaneously and continuously and even a single failure could cause catastrophe. Nowadays after a lot of research and experiments self-driving cars can be seen as a reality, or at least close future. Still, there are many challenges in designing a fully autonomous system for a self-driving car. The challenges of driverless cars can be sorted into five groups:

- **Road conditions:** Road conditions can be unpredictable and varying, i.e. in some cases they are smooth and well-marked while in other cases there are no lane marking. On some roads there can be potholes, in case of mountainous and tunnel roads the visibility of external signals for direction are poor.

- **Weather conditions:** It can be sunny and clear or rainy and stormy weather. Autonomous cars have to work in all weather conditions, there is no scope for any failure.
- **Traffic conditions:** Maybe this is the most complex problem as the human behaviour and autonomous car algorithm have to be matched with each other. The cars would have to drive in all sorts of traffic conditions as humans can do it. The people are emotional beings operating irrationally sometimes. Beside the fact that the traffic would be highly self-regulated, in certain cases some people might break the rules resulting an unpredictable situation in point of a self-driving car. An object may turn up in an unexpected conditions, e.g. crashed car which have to be bypassed ignoring some traffic rules. It can be handled by humans easily but for an artificial intelligence it is a harder task as its operation is ruled by rule-following conditions and this kind of situation needs the capability of detecting the need to ignore the rules in that specific case. The lack of intelligence of several autonomous cars on the roads could result in traffic deadlock.
- **Accident Liability:** One of the most important aspect of autonomous cars is accidents liability. Who is liable for accidents caused by a self-driving car? Generally, the software will be the main component that will drive the car and will make all the good and bad decisions. At initial levels of design there is a person who is physically behind the steering wheel and can take over the control, but in newer designs does not have even any dashboard, pedals and steering wheel. In such cars the driver, who is actually almost just a passenger cannot be the liable for an accident. Furthermore, due to the nature of autonomous cars, the drivers can be supposed to be in a relaxed state without paying sufficient attention to handle unexpected traffic conditions, thus by the time they need to act, it may be too late.
- **Radar Interference:** Autonomous cars use lasers and radar for navigation. The principle of radar works by detecting reflections of radio waves from surrounding objects. The time taken for the reflection is measured to calculate the distance between the car and the object and appropriate action is then taken based on the radar readings. The problems will come up when thousands of cars on the roads try to use the technology because the reflected radar signals can interfere making it difficult to distinguish which signal was emitted by which vehicle. Even if choosing a different radio frequency can eliminate the issue, the available frequency range is unlikely to be sufficient for all the vehicles manufactured. [5]

As a conclusion it can be stated that there are still lot of challenges beside the fact that there have been already some autonomous cars on the roads nowadays, but mostly for development and research purposes. Probably, the collective effort of the industry will definitely make the autonomous car in everyday life a reality one day because the benefits are considered to be remarkable. The most important expected benefit would be the reduction of losses of live in road accidents. Benefits with secondary importance are the reduction of fuel consumption, therefore the environment pollution caused by the vehicles, furthermore time and cost savings could be realized due to optimized behaviour. Time, cost and energy savings are top wanted purposes in almost every field of industry and everyday life, therefore autonomous solutions with such desired features will likely appear in more and more applications.

1.3 Application of autonomous transport robots

The conventional ways of transporting could be appropriate for most of the cases but not in those when repeated passages to specific locations in small distances have to be taken. One of the latest trends in technology is to create such driverless robots that can help people in their everyday life, i.e. bringing any kind of products to customers. A company called Starship launched a several self-driving robotic vehicles on the streets of Washington, D.C. that can transport food and other small items to customers ordering via app. The robotic vehicles move themselves along sidewalks using camera and tracking



Figure 1.2: Starship Technologies

technology to avoid any obstacle and its location can be tracked from the distance in order to know the time of arrival. The six-wheeled electric machine sits about a half-meter high having a lockable cargo bay to carry food products up to nine kilograms in weight.[7]

A company called SMP Robotics made automated guided vehicles that can remove grass clippings and fallen leaves or construct garbage collection in a garden.



Figure 1.3: SMP Robotics

These small-sized mobile robots have a trailer attached to the back. They are powered by electrical engines, thus noise and smell emission is minimized which allow for operation around people. The built-in accumulators stores the needed energy and can last for several days of cyclical operation. Two ways of route following methods are available. The first option is to choose a root in advance with the help of the attached PC tablet. The second way is to set the robot in its 'follow me' algorithm whereby it is going to follow the operator and learn the path. The collected goods on the trailer can be unloaded automatically using a mechanism without any human intervention. The trailer can be fitted with a water tank which can be used for watering of grass and other plants, furthermore automatic water refill algorithm is implemented. This kind of automatic self-moving watering system suits really well such occasions when a fixed irrigation systems are not worth being built due to climate characteristics.[6]

A firm named NEOBOTIX made autonomous transport systems for daily use in industrial applications. The robots are equipped with laser scanners that permanently detect landmarks and obstacles what allows the robot to react dynamically to unexpected changes of their surrounding, i.e. to safely operate between human and other moving objects. This flexibility makes them suited for dynamic transportation tasks with frequent changes. For instance, they can complement roller or belt conveyors and take parts to machines or workplaces that are not connected to the conveyor system. [8]



Figure 1.4: MT-400



Figure 1.5: MT-500

Possible applications are taking parts to and from workplaces, transporting in direct interaction with humans or dynamic picking of parts for later assembly

BMW logistics also uses autonomous transport robots similarly in industrial environment. The well-known car manufacturer has been working hard to reduce emissions in all steps of the manufacturing process of a car, not only in the final product. This application shows really well that the autonomous developments in the automotive industry is not just about the autonomous driving on the roads but the manufacturing processes as well. Smart Transport Robots (STR) transport components through logistics at the Wackersdorf plant. They measures the distance to wireless transmitters which are located at in the logistics hall to calculate its exact position and route. Using sensors to identify and react to critical situations, it is able to share the route with humans and other vehicles. [9]



Figure 1.6: BMW: Smart Transport Robots

A company, named AETHON, made fully autonomous robot called TUG that automates the transport of materials and supplies in commercial environments. It can

automatically drop off and pickup of carts, navigates on internal map of facility using laser and it has a omnidirectional 4 wheel driven drive system for higher maneuverability.



Figure 1.7: AETHON: TUG mobile transport robot

Another interesting field of application of transportation robots is the military. By putting humans to this work they are often exposed to a risk that could be avoided. The Autonomous Platform Demonstrator is a military transportation robot developed by the U.S. It has a hybrid-electric drive train with six in-hub electric motors powered by li-ion batteries charged using an on-board diesel generator.



Figure 1.8: Military transportation robot: APD

From control point of view it can be controlled in real-time by a soldier or it can operate autonomously. Autonomously it can operate at speeds up to 50 mph. Furthermore, it can travel along a GPS way point route and avoid obstacles in its way. [10]

2 Mobile Robots

The general platform of the transportation robots is obviously the family of mobile robots. Generally a mobile robot faces more design challenges compared to robots that operates in fixed positions as it is left alone in the world. The most critical design problem is related to the locomotion which has to be suited with the conditions of the given ground. In order to handle the changes of the environment and to be able to make decisions during runtime as function of actual state, the robots need to have more intelligence and self-determination. The mobile robots are still not reliable enough nowadays to operate in wide ranges outside of laboratory walls. They need to have a built-in model about their environment, furthermore they need to detect and analyse that of changes and their actual positions have to be found in order to be able to design and perform the next movements. In other words, a sequence of intelligent acts are expected to carry out based on interactions with the dynamically changing outside world.

2.1 Classification of mobile robots [2][1]

2.1.1 Based on the medium in which the robot moves

- **Air/Space**

These are mostly applied in space flight and military. The most well known application is the Unmanned Aerial Vehicle (UAV), commonly known as drone, which is an aircraft without pilot. The drones develop fast as they are easier to build and control as they do not have to navigate between obstacles.

- **Water**

Automated Underwater Vehicles (AUV) are generally used in seas and oceans for research and industrial purposes. One of the most interesting field is the development of amphibious vehicle that can operate either on earth or on water.

- **Earth**

As it is our natural habitat, the robots operating on earth are the most common.

2.1.2 Based on the locomotion mechanism

Mobile robots generally locomote either using wheeled mechanism, a well-known human technology for vehicles, or using a small number of articulated legs being inspired by biological systems. In general, legged locomotion requires higher degrees of freedom and therefore greater mechanical complexity than wheeled locomotion. Wheels, in addition to being simple, are extremely well suited to flat and hard ground, thus their energy need is radically less compared to legged locomotion. However, the softer the surface, the higher the rolling friction which results a dramatic efficiency loss in case of wheeled robots. The efficiency of wheeled locomotion depends on the flatness and hardness of the ground, while the efficiency of legged locomotion depends on the leg mass and body mass. The nature prefers the legged locomotion since they must operate on rough and unstructured terrain where a wheeled vehicle could get stuck easily, e.g. in case of Sojourner Mars rover in 1997. They are simply unable to step over an obstacle, the only chance is to avoid it. On the other hand, it is also understandable the most of the industrial applications of mobile robotics utilize some form of wheeled locomotion due to the fact that the human environment mostly consists of smooth surfaces, both indoors and outdoors. In such conditions, the wheeled configurations are currently faster and more mobile.

In locomotion, the environment is fixed and the robot moves by imparting force to the environment. The scientific basis is the study of actuators that generate interaction forces and mechanisms that implement the desired kinematic and dynamic properties. The core issues that are need to be evaluated to analyse a robot configuration in terms of locomotion are the followings:

- Stability
 - number and geometry of contact points
 - center of gravity
 - static/dynamic stability
 - inclination of terrain
- Characteristics of contact
 - contact point/path size and shape
 - angle of contact
 - friction
- Type of environment
 - structure
 - medium, (e.g. water, air, soft or hard ground)

2.2 Legged Mobile Robots

Legged mobile robots contact the ground by a set of contact points which provides good adaptability and maneuverability in rough terrains. Not even the ground quality is an issue until the ground clearance is possible to be maintained, e.g crossing holes which does not exceed the size of the robot is a solvable demand. The main disadvantages of legged systems include power and mechanical complexity. The leg must be capable of sustaining the part of the robot's total weight, lifting and lowering the robot body meanwhile keeping the stability permanently. Furthermore, high maneuverability needs sufficient number of degrees of freedom to impart forces in a sufficient number of different directions. The legged robots are basically biologically inspired based on the fact that they are successful locomotion systems. A number of different leg configurations have been successful in a variety of organisms, i.e. the most common numbers are the two, four or six.

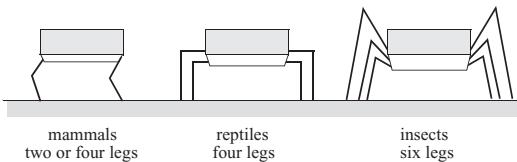


Figure 2.1: Arrangement of the legs of various animals

A creature with at least three legs can achieve static balance which means that no correcting movements have to be taken to preserve its balance. Actually static balance means that the center of gravity is within the tripod of ground contact. A small deviation from stability is passively corrected toward the stable pose by constrain forces after the

disturbance goes away. However, any creature who is intended to walk must be able to lift and then release their legs, thus for static walking the minimum number of needed legs is six. Insects and spiders are immediately able to walk when born compared to mammals with four legs who are not able to walk or even to stand in case of humans with only two legs. Therefore the main issue regarding the locomotion of legged robots is the balancing. Of course, the number of legs and also the number of degrees of freedom of a given leg can be increased in order to achieve better stability and maneuverability. On the other hand, more legs and joints result more actuators, thus more energy, more mass, more control, i.e. more costs in certain senses. Most of the legged robots in research and industrial cases have two legs. The robot legs and arms are usually driven by electrical motors and pneumatic pistons. High level of harmony of the piston movements must be carried out by the designer in order to its hold balance also during the walking what is challenging control task.

2.3 Wheeled Mobile Robots

The wheels are the most widely used locomotion mechanisms in man-made vehicles in general due to its simplicity, effectiveness. In most of the cases, during the design the robot builders need to focus problems regarding traction, stability, maneuverability and control.

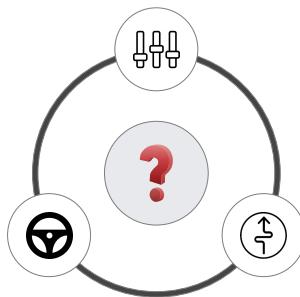


Figure 2.2: Major aspects of wheeled robot design

In general, balance is not usually an issue in wheeled robotics as they are designed such a way that the all wheels are in contact with the ground. Having three wheels is sufficient to guarantee the static stability if the center of gravity is within the triangle formed by three contact points. Obviously in case of two wheels, self-balancing is necessary. When more than two wheels are used, once the stability improved by adding more contact possibilities, but also a flexible suspension system is required to allow all wheels to maintain ground contact when the robot encounters uneven terrain. One of the simplest approaches to suspension is to design flexibility into the wheel itself, i.e. applying deformable tire of soft rubber material.

Basically, the maneuverability of a given robot depend on the design, the configuration and the degrees of freedoms of powered wheels. Each configuration, wheels type has their advantages and disadvantages that have to be considered before building a mobile robot, we shall see the details later. The robot arrangements can be classified into two groups: holonomic and anholonomic. A mechanical system is called anholonomic if in a specific configuration (state) there exists such displacement or rotation that cannot be carried out by means of the combination of possible actuators. A well known example, an automobile cannot move perpendicular to its orientation because of the kinematic constrains of the wheels. On the other hand, some robots are holonomic, in other words omnidirectional, meaning that they can move at any time in any direction along the

ground plane (x,y) regardless of the orientation of the robot around its vertical axis. This level of maneuverability requires wheels that can move in more than just one direction, we shall see the details later too.

There is generally an inverse correlation between controllability and maneuverability. For example, the omnidirectional designs require significant processing to convert desired rotational and translational velocities to individual wheel commands. Furthermore, such omnidirectional designs often have greater degrees of freedom at the wheel. Controlling an omnidirectional robot for a specific direction of travel is also more difficult and often less accurate when compared to less maneuverable designs.

As conclusion, there is no perfect robot configuration that simultaneously maximizes stability, maneuverability, and controllability. For that reason, it is always the designer's task to choose the most appropriate mobile robot configuration considering the constraints originated from the given application.

2.3.1 Wheel design

The choice of the wheel type has a large effect on the overall kinematics. There are four major wheel classes as shown in the figure below. The standard and castor wheel

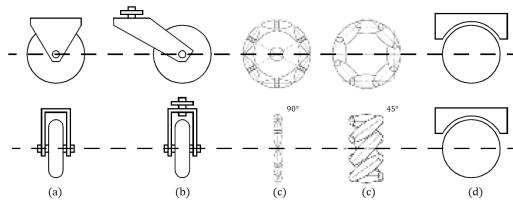


Figure 2.3: Major classes of wheels: (a) Standard wheel, (b) Castor wheel, (c) Swedish wheel (45°/90°), (d) Ball wheel

have are highly one-directional as they have a primary axis around they can rotate. These wheels must be steered along a vertical axes to move in a different direction. The main difference between the two configuration is that the castor wheel rotates around an offset vertical axes causing adverse force imparted to the case compared to the standard wheel case where the axes of rotation passes through the contact point. Both the Swedish and the spherical wheel are less directionally constrained than the former conventional ones. The Swedish wheels functions such as normal wheels providing low resistance in another direction as well, i.e. rolling resistance instead of sticking. Small rollers are attached around the circumference can rotate only in a passive way while the primary axes are the only powered joints. As a result, although the wheel can be driven along only one principal axis, the wheel can move along almost any trajectory. The spherical wheel is a truly omnidirectional wheel, often designed so that it may be actively powered to spin along any direction.

2.3.2 Wheel configurations

2.3.2.1 Holonomic robots

In practical applications, holonomic systems are rarely present due to higher prices and mechanical complexity, moreover dead reckoning by means of odometry is quite inaccurate.

Three wheeled omnidirectional mobile robot

The mechanically simplest holonomic robot is omnidirectional configuration. At least

three independently driven wheels are needed to be able to move in any direction (x, y, φ) in ground plane at any time. The resultant velocity vectors in terms of wheel velocities

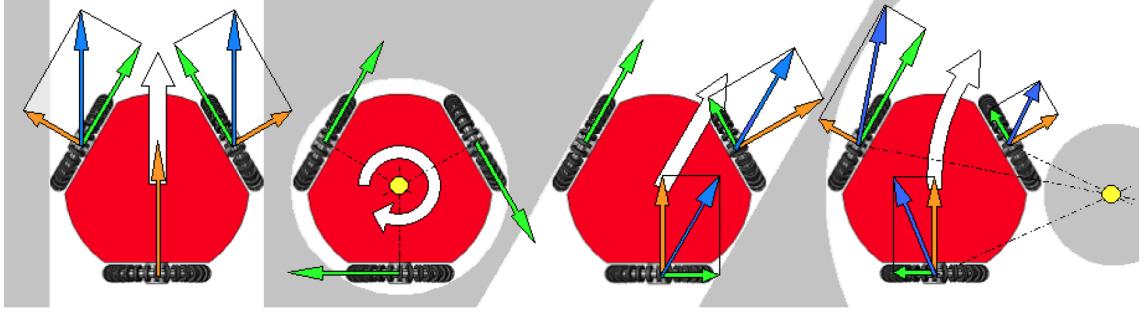


Figure 2.4: Three-wheeled omnidirectional drive configuration

can be seen in Figure 2.4. The blue wheel velocity vectors can have lateral component, i.e. the wheels have to be able to move parallel to their primary axes. This kind of configuration needs the previously mentioned Swedish wheels or spherical wheels. In addition, also four wheels or even more can be built-in in a system, but in this case it can occur that the wheels drive each other in reverse direction, furthermore the more than three contact point does not guarantee the proper contact of each wheels with the ground.

Omnidirectional robot with four Swedish wheel

Four-wheeled arrangement with 45° Swedish wheels, each driven by a separate motor, have been used in several research and industrial projects. A good example is is Uranus that can rotate and translate independently without any constraints by means of omnidirectional wheels.



Figure 2.5: Carnegie Mellon Uranus robot



Figure 2.6: Mecanum AB robot

When all four wheels spin in the same direction, the robot moves longitudinally in a straight line forward or backward, accordingly. When one diagonal pair of wheels spin in the same direction and the other diagonal pair of wheels spin in the opposite direction, the robot moves purely laterally. The simultaneous spin can be achieved if the wheels on left side are driven in forward or backward while the wheels on the other side are driven the corresponding opposite direction. As it was discussed earlier, this four-wheel arrangement is not minimal in terms of control, stability and maneuverability, but preferred over three-wheeled configuration for higher capacity and stability reasons.

Main disadvantages of omnidirectional configurations includes the complexity of control because of the more complex processing of wheel velocities. For instance, the Swedish wheels has a set of free rollers along the wheel perimeter adding extra degrees of freedom,i.e. translation in the lateral direction. Furthermore, due to these rollers the wheel perimeter is not a perfect circle, thus accuracy of dead-reckoning based on counting the number of rotations might be inaccurate soon. The rollers can also cause slippage, thus accumulation of displacement errors.

2.3.2.2 Anholonomic robots

The non holonomic wheeled robot configurations are more widely spreaded, e.g. automatic wheelchairs, robot cars. The most common choice of indoor robotics is the well-known two-wheel differential drive due to its simplicity beside acceptable maneuverability, only slightly inferior to omnidirectional drives. In optimal configuration the two wheels are arranged symmetrically, thus the robot can spin around the center point of the case without changing its location. One or two additional ground points, e.g. non driven castor wheel(s), are usually added to gain stability. In such robotics, motion in a partic-

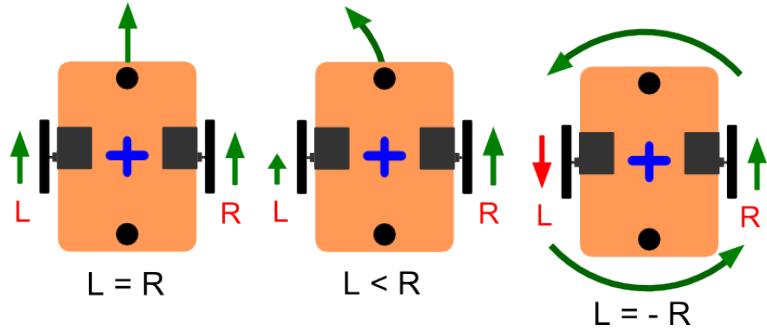


Figure 2.7: Two-wheeled differential drive configuration

ular direction may initially require a rotational motion to be able to change the position along straight movements. Two wheel differential drive is not clearly omnidirectional but motion can be decomposed into distinct rotation and displacement according to the given order. One can say, that it has two and a half degrees of freedom compared to three wheeled omnidirectional configurations that have three whole degrees of freedom. The advantages includes easy maneuverability, simple construction as only two motors and two conventional wheels needed. Also two wheels can be enough, but in this case self-balancing control has to be attached, in general castor wheels are preferred. The disadvantages of this configuration are lower lateral stability at higher speed turns, sensitivity to terrain roughness, furthermore if the axes of the two wheels are not coincident, the sliding of the wheels will occur.

Similarly to automobiles, remarkable part of the four-wheel robots use the Ackerman steering geometry. It is a way of geometric arrangement of linkages in the steering of a vehicle designed such a way that the inner and outer wheel of a turn can move along circles of different curvatures. It is important that θ_i and θ_o angles cannot be the same

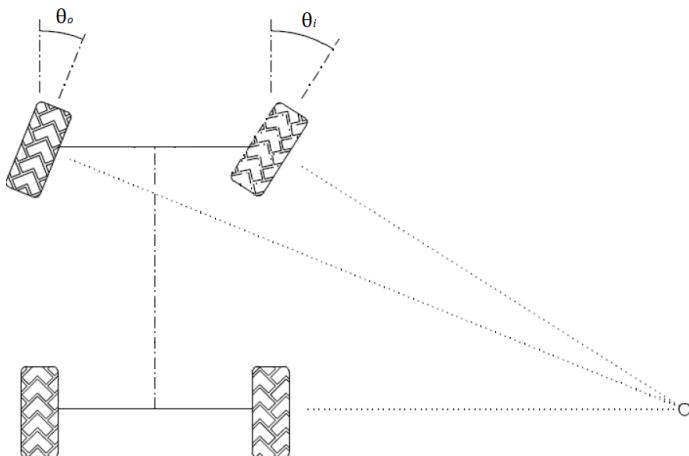


Figure 2.8: Ackerman steering arrangement

whereas in that case the constant sliding of the wheels would be inevitable. In addition, the rear wheels also have to be tuned to spin with different appropriate velocities as they move along different trajectories. As Figure 2.8 depicts, such a vehicle typically has a turning diameter that is larger than itself, thus to move sideways requires a parking maneuver consisting of repeated changes in direction forward and backward. In addition, the limited maneuverability of Ackerman steering has an important advantage, i.e. its directionality and steering geometry provide a very good lateral stability in high-speed turns. Furthermore, it is much more easier to straighten simply by locking the steerable wheels and driving the drive wheels compared to the omnidirectional configurations where permanent control is needed.

Multi-degree-of-freedom robots have more than the needed three degrees of freedom. They have enhanced maneuverability, but as a result of the over determined arrangement the slippage free control is more challenging at the same time. The main application scope are mostly the navigation on uneven terrains, e.g. military applications or moon rovers. The position tracking is impossible by odometry, thus external sensors are essentially needed.



Figure 2.9: UNIQ: 8 DOF mobile robot



Figure 2.10: Sojourner: Mars rover of NASA

3 The Mobile Robot

3.1 Selecting a mobile robot configuration

In previous chapters the basic features of mobile robots and some real life application were presented. This final project basically aims at designing an autonomous transport robot to investigate its control and possible parameter estimation problems. Real world transport robots are usually capable to navigate along prescribed path, monitor their environment to handle unexpected obstacles, interact with the packages autonomously and communicate with users surrounding them. These are the most necessary features that are really needed during operation which have been mostly implemented in the past.

The development of kinematic and dynamic properties have been taken less attention so far, therefore this field can contain remarkable development potential in my opinion. The transport robots in industrial application are designed such a way that during their movement they will not likely face issues regarding the stability because they move along high radius path with low speed and accelerate moderately. In other words, a compromise is made between the configuration costs and the productivity, between the security and effectiveness. Sooner or later, the demand for a more dynamic, faster and smoother operation of transport robots can come into being just like in automotive industry. The problem can be more complex as well in certain situations. For instance, a transport robot operating in a plant that carries loads with different properties such as mass, moment of inertia or height would face a more challenging situation in dynamic stability point of view than a car with more or less known and constant physical parameters.

To select a mobile configuration, the general purpose, working environment and concept of the robot have to be defined. Based on real life experience and possible future application needs, this study will focus on designing a transport robot which would operate in a plant, in a laboratory or on the streets carrying different shaped and weighted loads. Instead focusing on navigation, i.e. dead-reckoning, route planning and obstacle avoiding, the control of dynamic behaviour is to be examined in details. Beside the technological aspects also the practibility and costs have to be considered because the building of the mobile robot configuration is also the part of my thesis. The most important aspect of a study of such a advanced moving transport robot is the maneuverability. In terms of maneuverability, the omnidirectional robots show the best performance. Choosing a three-wheeled omnidirectional mobile robot configuration can be regarded the optimal choice. The controllability is more complex compared to anholonomic robots, but with the choice of minimum number of three wheels it is still a handleable design challenge. Application of more wheels could give more controllability which would be favoured, e.g. TUG is also four wheel driven omnidirectional configuration. However, it would make some additional issues appear such increased slip danger and more complex controllability. Compared to anholonomic configuration, the holonomic configurations are less stable as mentioned before, but it can be regarded as an advantage in this case, because this study is willing to deal with stability issues, thus the main objective is to examine the behaviour of the system near stability and provide parameter estimation methods to avoid it. In other words, a transport robot might be preferred over others if it can be as stable as possible without loosing any maneuverability skill. For an omnidirectional drive configuration special wheels of higher price are need, thus the three-wheeled configuration is optimal from economic point of view.

3.2 The Hardware

The realised mobile robot was built from scratch ordering its part from different sources. In this chapter, the components of the mobile robot platform are presented.

3.2.1 Omni wheel

In order to fulfill, the maneuverability demand a configuration with omni wheels was looked for. As it was mentioned, the omni wheels are like regular ones which can roll around their axis but with additional passive rollers along its circumference which allows the wheel itself to move in the lateral direction as well. Because of the additional rollers, the contact point(s) (mainly there are two of them even if rigid wheels are considered) will not be always under the wheel center, e.g. when shifting between rollers occurs. As a result of that, the movement on flat surfaces will be less smooth compared to conventional wheels. Applying more rows of rollers, the uneven rolling characteristic (transition between passive rollers) can be improved. However, the contact point changes in the lateral direction too. [26] The ordered wheel is shown in the Figure 3.1. Such a wheel is made

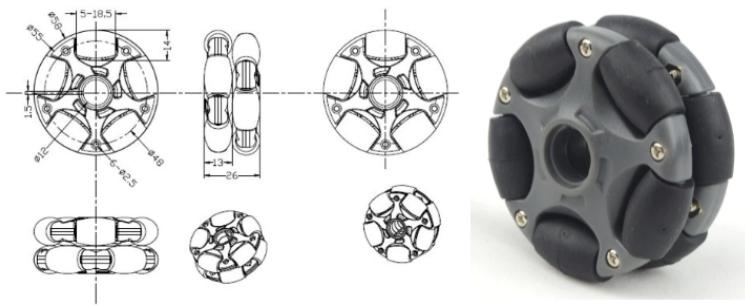


Figure 3.1: Omni wheel construction

of plastic having maximum nominal load capacity of 3kg and 58mm diameter. It can be connected to the shaft of a DC Motor using a coupling hub and screws.

3.2.2 Gear Motor

In mobile robotics, the DC (direct current) motor is a general choice for actuation purposes. The points which are to be considered are as follows:

- Its nominal operating voltage and the current consumption which is needed during the possible operating states. The energy source and the electronic devices have to be able to provide the electrical energy in the given needed form.
- Motor torque has to be in match with the inertias of the robot configuration.
- Motor speed has to be in match with the demands regarding the demanded locomotion speed of the case.

At the given scale, the DC motors are more commonly brushed permanent magnet type. They runs at quite high speed about 10,000-12,000 RPM having a little torque.[20] For a robot application, it is too fast and the torque is too little, therefore motors with gearbox attachment are needed. The gearbox is a set of gears containing several stages to increase the motor torque at the cost of the shaft speed. The selected DC gear motors are represented in the Figure 3.2.

DC Gear Motor Datasheet

Rated voltage	DC 12.0 V
No-load speed parameter	107 [RPM]- 0.15 [A]
Maximum efficiency point parameter	85 [RPM] - 0.6 [A] - 0.42 [Nm] - 3.8 [W]
Maximum power point parameter	52 [RPM] - 1.2 [A] - 0.98 [Nm] - 6.0 [W]
Blocking point parameter	3.4 [A] - 1.96 [Nm]
Gear ratio	1:90
Encoder resolution	1:11

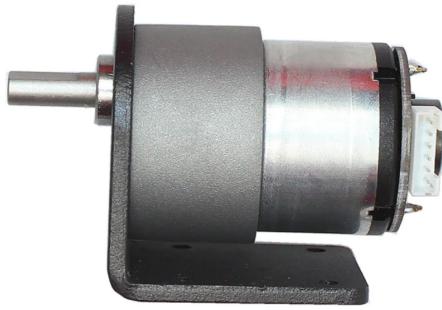


Figure 3.2: Selected gear motor

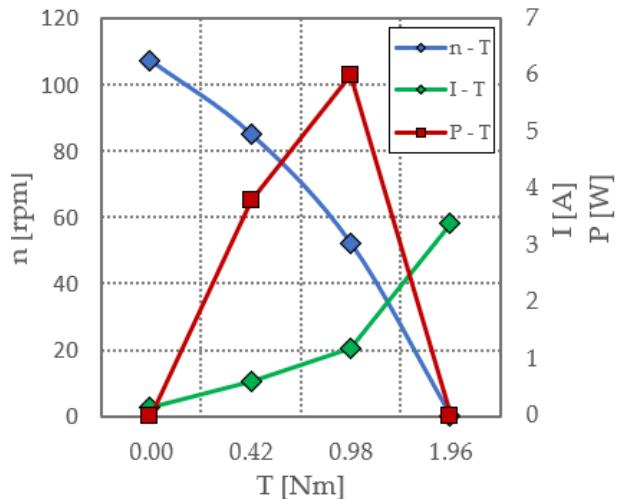


Figure 3.3: Gear motor characteristics

The supplier provided the nominal values of specific working points which are shown in the Figure 3.3 and listed in the table above. The motor is equipped by magnetic quadrature encoders in order to allow to count the revolutions. There are two Hall-effect sensor which are able to detect the changes of the magnetic field. If a gear teeth passes near in front of the sensor, the magnetic field will be changed which will cause the output of the sensor to either go high or low. The frequency of the generated square wave will be proportional with rotational speed of the shaft. In order to identify the direction of the rotation (clockwise or counter-clockwise) a second sensor is attached to motor. Eventually, the encoder provides $2 \times 2 \times 11 = 44$ counts per revolution considering that the observed gear has 11 tooth, two hall sensors are present and both rising and falling edge are counted. In addition, the motor has a gearbox with a ratio of 1:90, thus the Hall-effect sensor generated $44 \times 90 = 3960$ counts per revolution. Therefore the accuracy in theory is about 0.09° .

3.2.3 Case

In Figure 3.4 a photo of the assembled robot case is shown. The DC motors are attached to the case using L shaped brackets. One, two or three levels can be built from triangle shaped metal sheets which have a black electrically insulating coating. The main purpose of the frame beside making connection between the wheels is to provide a safe place for all components of the robot, such as the electronic modules. Using the center free hole in the most upper level, a threaded rod can be attached to the case to provide a place for mounting additional payloads demonstrating the weight of an arbitrary package. The side size is 145mm, thus common wheel circle has a radius of 0.12m. The weight of the case, motors and the wheels is about 1.3kg.



Figure 3.4: Robot case (bottom)

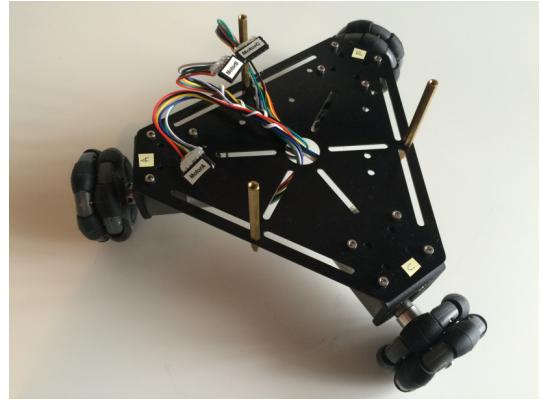


Figure 3.5: Robot case (top)

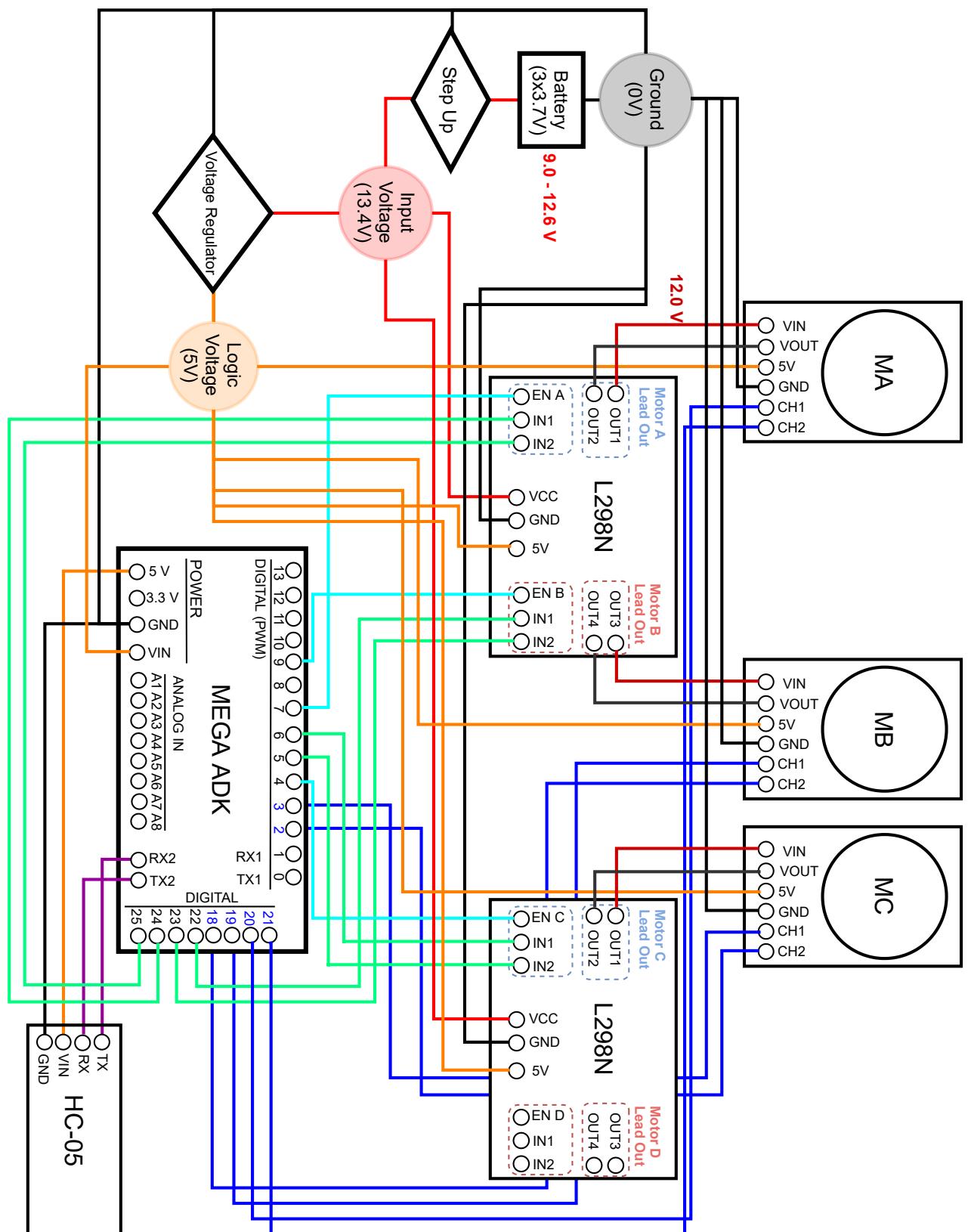
3.2.4 Electronics

A RobotDyn Mega 2560 board with an ATmega2560 (16Mhz) microcontroller was chosen which is a cheap version of the original Arduino board having the same capabilities and compatibilities. Arduino is an open-source electronics platform including easy-to-use hardware and software.[27] The programming language is based on C++, the Arduino library is well-documented, widely used and easy to learn. There are several boards compatible with Arduino at lower prices as well. The Mega actually was chosen for its increased number of pins compared to base models such as Uno. Considering the rated voltage of the DC motors and other aspects such as space and performance, 3-celled LiPo (litium polimer) with 3x3.7V rated voltage battery was chosen to power the mobile robot. This type is widely used for its good capacity over weight ratio, high current supply capability and fast charging. The main disadvantage of this kind of batteries that they can flame up in case of short circuit caused by a serious mechanical damage or some kind of unnormal use. For safety reason, an electrical fuse was added to protect the battery and also the electronic circuits from operating under overcurrent. Generally, two voltage levels have to be provided: a logic circuit with 5V and a power circuit that provides 12V at motor pins, in addition, a common ground has to be formed. This separation is essentially need since the microcontroller board works with an operating voltage of 5V and also other electronic circuits can operate on that voltage level, furthermore the board can provide maximum of 40mA at a single pin which would not be sufficient to drive the motors. The board has an own voltage regulator but it is recommended to power with a voltage less than 12V, thus for driving the motor a higer voltage circuit is essential. [27] A DC motor can be switched on/off using a MOSFET transistor connecting a digital output of the board to its gate and providing high or low voltage, thus opening or closing it. The technique makes it possible to connect higher voltage and current to the motor directly from the power source using a programmable electronic device. If constant voltage is supplied, a DC motor will spin as fast as it can in accordance with inner and outer resistance type load which characteristic is shown in Figure 3.5. The most efficient way to control the speed of a permanent magnet type DC motor is using the PWM technique. The pulse-width modulation (PWM) technique gives an alternating voltage signal with a given period of time which can be only zero or high voltage values. The percentage of the time period when the voltage level is high called the duty cycle. If a pwm signal with a sufficiently high frequency is generated, the motor with a much larger time constant will sense as if it was a direct voltage input signal with a given average voltage value. This is because of the fact that a motor has inertias (J, L) and frictional losses, therefore it takes time to react to sudden changes and it will filter the PWM command resulting in an

averaged signal.[18] [19] The average voltage value will be proportional with the duty cycle of the PWM signal. Such an alternating signal, e.g. a square-wave can be generated with some of the digital output pins of the arduino board. Unfortunately, for controlling the direction of rotation speed further components are needed. The motor will spin in the reverse direction if the polarity of the voltage, thus direction of the armature current is switched. A general solution for that is to applying H-Bridge circuit which contains four switches, e.g. transistors, and the motor itself. By working the appropriate switches, the direction of the current flow can be changed. [14] The so called motor divers have these functionalities, in my case the usually used L298N driver was chosen as it is cheap and can do what is needed. The main disadvantage of such drives that they cause a substantial voltage drop, i.e. 1-2V in case of 12V supply voltage which will result in a lower motor speed. To handle this problem, an additional step-up converter was attached just after the battery to have a higher supply voltage. After some trial and error method the supply voltage it was found that if the supply voltage is stepped up to 13.4V, then the motors will get 12V at maximum duty cycle of the pwm signal. With this extra step up module, also the voltage drop of the battery (when it goes dead) can be handled and kept on a constant voltage with a higher supply current. The logic voltage high level (5V) is produced by a voltage regulator connected after the step-up module. The logic voltage levels have to be wire into all the devices, thus the motor encoders, microcontroller unit (MCU) and motor drivers (L298N) together with the common ground. The motor driver has to be connected also with the input voltage. The pwm signals are provided by specific pwm capable output pins of the board are to be connected to the Enable pins of the drivers. Additionally, two digital pins have to be connected to the drivers which control the rotational direction of the motors. The motor encoders has two pins called CH1,CH2 which have to be connected to the special interrupt capable pins of the MCU. The interrupt technique guarantees to catch all the pulsed generated by the encoders. If a rising or a falling edge is sensed, then the main loop is interrupted while a predefined so called Interrupt Service Routine (ISR) is called and processed.

In order to be able to communicate wireless with robot, a bluetooth module (HC-05) was used by which serial communication could be realised. The serial communication is a process of sending data one bit at a time sequentially over a communication channel. For instance, the Arduino board can communicate with the computer via an USB type cable using the serial communication protocol, e.g. when the code is uploading or when commands are sent from the board to the pc or vice versa. The pins for serial communication are denoted as RX (Receiving) and TX(Transferring) letters, the USB connection is connected to RX1,TX1 serial channel on the Mega. In order to avoid the collision of communication of the bluetooth module and the USB port if both are in use at same time, the BT module was connected to the RX2,TX2 to channel.

3.2.5 Schematic diagram of the electronics



3.3 DC motor speed analysis

3.3.1 System modelling

The robot configuration is equipped by three officially identical 12V DC motors with permanent magnets having the nominal rotational speed value of 107RPM at no load state. They have to be driven synchronously in order to obtain the desired movements. During operation the demand for having a variable motor speed is essential, thus control algorithm has to be developed. Briefly, the DC motor works on the principle (Lorentz Law) that when a current conductor wire (armature) is put in a magnetic field then it will be subjected to electromagnetic force in accordance with the direction and magnitude of current and magnetic field, i.e. $\mathbf{F} = I \mathbf{l} \times \mathbf{B}$. The coil is connected to a DC power source, thus current will flow through the wire producing its own magnetic field. The mechanical force is produced as result of the magnetic field of the permanent magnets and the magnetic field of the coil. The magnetic field of the coil is attracted to the permanent magnetic field, therefore the armature starts rotating due to the induced electromagnetic force. At some point the magnetic fields would align resulting in a zero torque. In

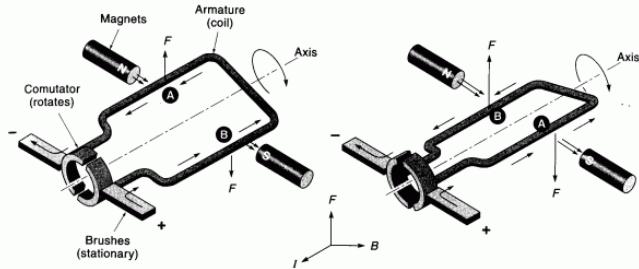


Figure 3.6: A single current conductor in a static magnetic field [17]

brushed DC motors this switching mechanism is done by the so called commutator part. The Figure 3.6 represent a single wire with a commutator. The commutator is a simple mechanical switch that changes the polarity of the voltage put on the wire by touching the brushes in order to make the current flow in such a direction that maintains the direction of the generated torque, therefore the armature continue rotating. A single wire would provide a fluctuating torque during the operation since the angle between the coil and static magnetic flux is changing. For a more smooth operation of the rotor, several separate wire loops have to be used shifted from each other and switched one after the other. This can be realized by applying more pairs of commutators which enables to connect always that coil loop with power source that does not align with the permanent magnetic field, thus can produce maximum torque. In practical motors the rotating part is called armature and standing part is called stator. The stator can be a permanent magnetic pole in case of small DC motors, while in larger motors an electromagnet is used which is also powered by the same DC source in serial or parallel connection. In my case, the motor has a permanent magnetic stator. According to Faraday's law of electromagnetic induction, a rotating loop of coil will produce an electromotive force called EMF. The case of the rotating armature loop is the same, thus in this case it produces an adverse electromotive force which is called back EMF that tends to decrease cause of the accelerating effect, i.e. the applied input voltage. As a result, the back EMF will decrease the armature current by a remarkable amount, thus the motor torque as well. [16] [17]

The electrical equivalent circuit of the armature and the free-body diagram of the rotor can be modeled as follows.

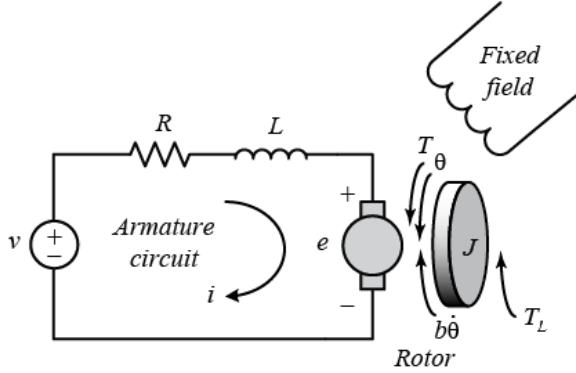


Figure 3.7: DC motor electrical and mechanical model [18]

The motor is supplied with V voltage as an input, while the output is the rotational speed of the rotor and shaft (rigid connection). In the mechanical model the load torque (T_L) and also dissipation energy is considered as being proportional with the angular speed. The motor torque is generally proportional to the armature current and the strength of the magnetic field, but in this model the magnetic field is assumed to be constant, thus the motor torque depends on the armature current (i) only. The back emf (e) is proportional to the angular velocity of the shaft. If SI units are used then the constants of the torque and the emf are equal, thus the same (K) constant can be used for the following equations:[18][19]

$$T = Ki \quad (3.1)$$

$$e = K\dot{\theta} \quad (3.2)$$

Based on the model in Figure 3.7 the following equations can be derived by means of the Newtons's 2nd Law and Kirchoff's voltage law.

$$J\ddot{\theta} + b\dot{\theta} = Ki - T_L \quad (3.3)$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \quad (3.4)$$

The system can be described by a system of linear differential equations with respect to armature current and rotor angular velocity. A given system can be defined by 5 parameters, i.e. armature resistance (R), armature inductance (L), motor constant (K), damping constant (b) and motor inertia (J).

3.3.2 Step response analysis of the motors

The motors speeds can be changed via PWM and with the help of the rotary encoders the response function to any voltage value could be measured. It is more convenient to treat the PWM signal's duty cycle as the control input although voltage level is applied to the armature of the motor. The PWM signal's duty cycle is determined in the code by a floating point number between zero and one. Firstly, let the pwm value be regarded as an input and the wheel speed as an output assuming that the transfer function of the gearbox and power electronic module are proportional type, thus it can be handled together with the motor as a plant element. The motors gave a quite similar responses with respect to time constant and amplitude and also the nominal value (107RPM at 12V no load case) was obtained with small deviations.

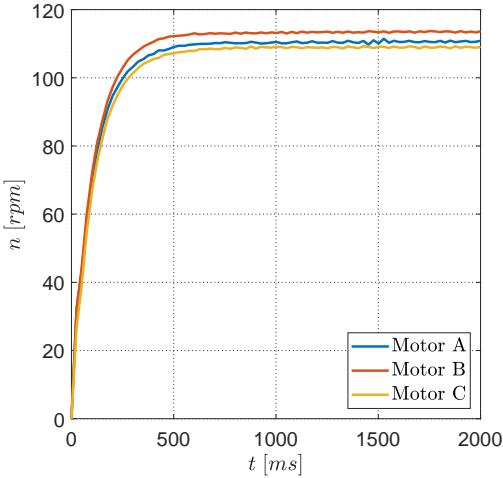


Figure 3.8: PWM=100% step response of the motors

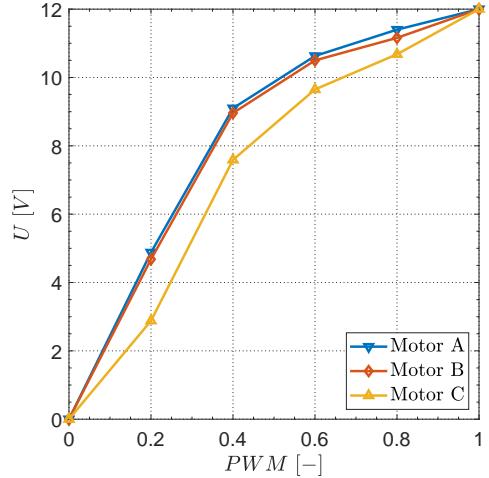


Figure 3.9: Nonlinearity of the PWM-U relation

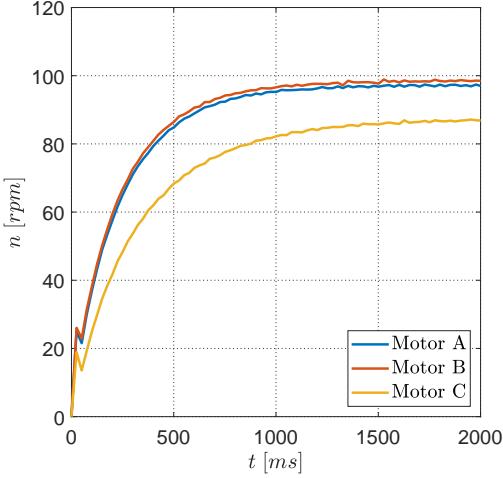


Figure 3.10: PWM=60% step response of the motors

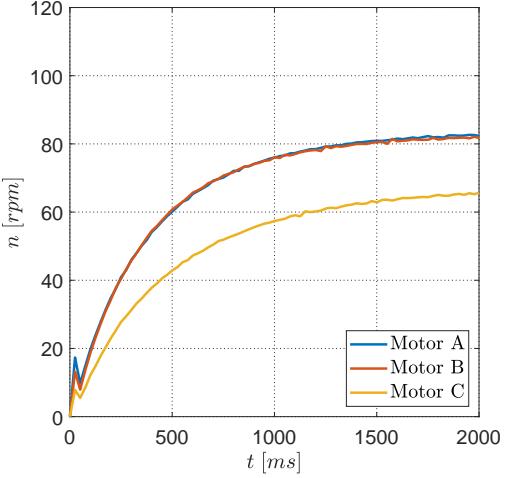


Figure 3.11: PWM=40% step response of the motors

Unfortunately, the problems were captured when the duty cycle of the pwm signal was decreased, i.e. the pwm value in the Arduino code. The prior guess that the pwm value provided in the MCU code will be proportional to the motor speeds was proven to be wrong. Furthermore one of the L298N module was functioning with a different characteristic as the other driver as it provided different voltage values for the same pwm signal which resulted that the Motor C was spinning remarkably slower. It is also clearly visible that the system is nonlinear as neither for the Motor A and B the same dynamics with respect to rise time were obtained, thus even the time constants were not similar. It is obvious that without proper control of the wheels it is hard to achieve smooth and accurate movements which is one of our main design criteria. As a first approximation, it is worth neglecting the mentioned issues and handle the system as it would be completely linear. This is defenietly not true but a good apprcimation. Further problems could occur at low voltages, i.e. pwm values less than 0.2 where the system probably become extremely nonlinear as the motor will not move at such small voltages where the generated motor torque is not strong enough to overcome sticking friction. Though the linear viscous

friction is included in the model presented in the next chapter as it is easy to calculate with. A possible practical solution is to have the motor banned to work in the problematic regions.

3.3.3 Controller design using blackbox model [18]

Considering the system as being linear in theory based on the response function corresponding to any pwm signal, a controller can be designed for the whole range for each wheel separately. In order to do this, a black box model has to be constructed. In case of a DC motor it is a typical procedure to handle the system as a first order one. [18] Applying the Laplace transform on Eq.3.3 and 3.4, then rearranging the transformed equations the transfer function of the system is as follows:

$$G(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}. \quad (3.5)$$

Beside the fact that a second-order model can be derived, the step response of dc motors usually looks like that of a first-order system. After inspecting the actual step response of the given motors, it looks as a good approximation. It can be explained by the fact that the motor is an overdamped second-order system, in other words it has two negative poles (roots of the denominator of the transfer function). Such a system is stable and has two time constants. In case of DC motors, the time constant of the mechanical dynamics is much higher compared to the time constant of the electrodynamics, thus the slower mechanical dynamics dominates the overall time response. A first order system can be expressed as it follows:

$$P(s) = \frac{\dot{\theta}(s)}{DC(s)} = \frac{k}{1 + \tau s}, \quad (3.6)$$

where

- $DC(s)$ - Laplace tranform of the duty cycle signal, $P(s)$ - plant tranfer function
- τ - time constant which equals the time it takes the system to reach the 63.2% of steady-state step response of a unit step, more generally it represents the time scale of the system dynamics,
- k - gain, the ratio of the magnitude of the steady-state step response to the magnitude of the step input.

Based on the measurments shown in Figure 3.8 the following values were calculated providing a really good agreement. A common choice for such a system is a PI controller.[18]

	A	B	C
k [rpm]	111.05	113.39	109.07
τ [ms]	100.70	99.88	107.03

Table 3.1: First-order model parameter estimations

The plant block is overall model of the power electronic module, the motor armature, rotary encoders and the signal processing on the MCU. The transfer function of the controller PI (proportional-integrating) is as follows:

$$C(s) = K_p + \frac{K_i}{s}. \quad (3.7)$$

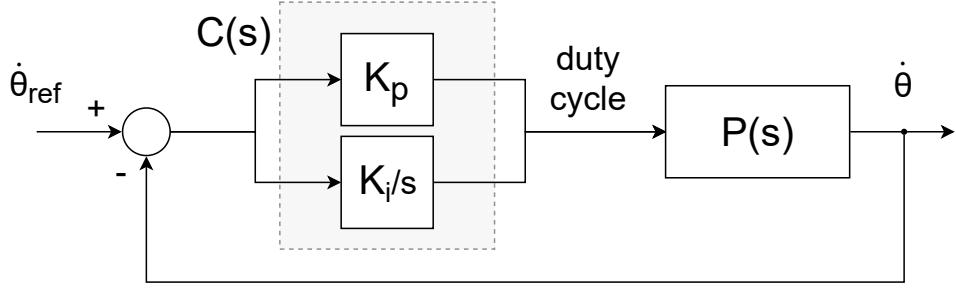


Figure 3.12: Model of the closed (feedback controlled) system

The transfer function of the (negative feedback) closed-loop:

$$T(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{A(K_i + K_p s)}{AK_i + AK_p s + s^2\tau + s} \quad (3.8)$$

The closed-loop system is almost a second-order one except for the presence of the zero (possible root in the denominator), thus it does not match with the canonical form exactly:

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.9)$$

where

- k - gain is the ratio of the magnitude of the steady-state step response to the magnitude of the step input,
- ζ - The damping ratio is a dimensionless quantity characterizing the rate at which an oscillation in the system's response decays due to effects such as viscous friction or electrical resistance,
- ω_n - The natural frequency is the angular frequency that the system will oscillate at when there is no damping. [18]

The original system is overdamped but the addition of a proportional controller can make it underdamped as it has similar behaviour like a "virtual spring". For such special case several design equations are available if the parameters in the canonical form are known. The presence of the zero in closed-loop transfer function can be a cause of errors. However, as a first guess it can provide a good approximation for starting value for the controller if the presence of zero is not considered and the system is treated as if it was an underdamped second-order system. It is important to mention, that the closed-loop system would have a canonical form if only proportional type controller was used.

The canonical second-order transfer function has two poles at:

$$s_p = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} = -\sigma \pm j\omega_d. \quad (3.10)$$

If $\zeta < 1$, then the system is underdamped, the two poles are both complex-valued with negative real parts. The system is asymptotically stable and oscillating with the damped natural frequency $\omega_d = \omega_n\sqrt{1 - \zeta^2}$. Based on the value of damping the system can be marginally stable and unstable, but those are not the cases which are have to be examined now.

Combining the denominators of equation (3.8) and (3.9) the following expressions can be obtained:

$$\omega_n^2 = \frac{AK_i}{\tau}, \quad (3.11)$$

$$2\omega_n\zeta = \frac{1 + AK_p}{\tau}. \quad (3.12)$$

Additionally using the equation 3.10, the relation between the poles of the closed loop and the controller parameters can be derived:

$$\sigma^2 + \omega_d^2 = \frac{AK_i}{\tau}, \quad (3.13)$$

$$\sigma = \frac{1 + AK_p}{2\tau}. \quad (3.14)$$

The figure 3.13 represents the step response of a canonical second order system and several properties are denoted to characterize itself.

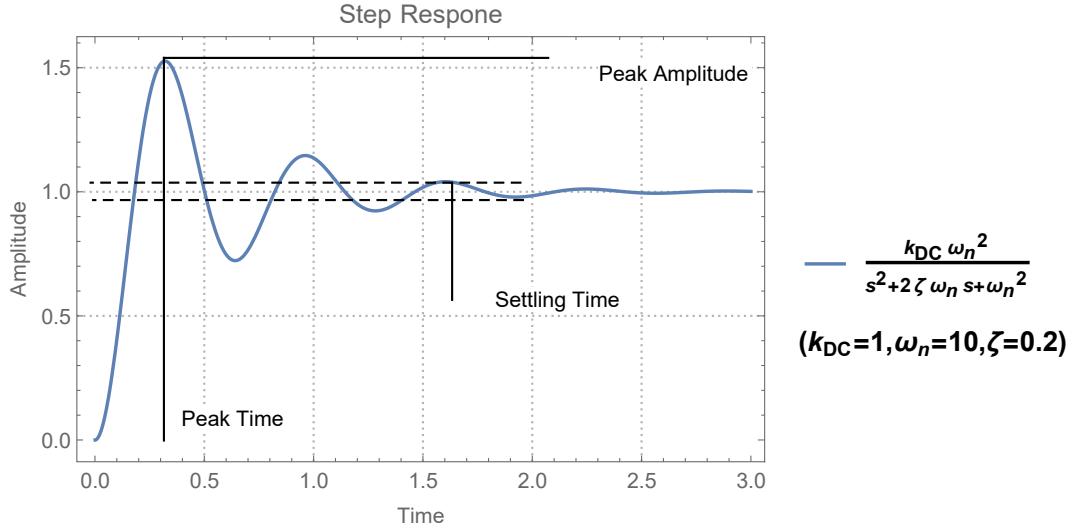


Figure 3.13: Step response characteristics

- Settling Time is the time that the system needs to have an output with a smaller deviation from the steady-state in terms of a given percentage. In case of second-order canonical system the following formula provides a good approximation: [18]

$$T_s = -\frac{\ln(\text{tolerance})}{\zeta\omega_n} = -\frac{\ln(\text{tolerance})}{\sigma} \quad (3.15)$$

- Percent Overshoot characterize the output in terms of the ratio of maximum peak and the steady-state value expressed as a percentage. [18]

$$Mp = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (3.16)$$

The corresponding peak time is as follows: [18]

$$T_p = \frac{\pi}{\omega_d} \quad (3.17)$$

Ideally closed-loop respond with low settling time and overshoot is wanted. Limits can be defined in arbitrary way that suits are demands with respect to the former properties. Based on equations 3.15, 3.16 and 3.17 further limits can be derived for ζ and ω_d .

- 1 % settling time:

$$T_s = -\frac{\ln(0.01)}{\sigma} = \frac{4.605}{\sigma} < l_{T_s}, \quad (3.18)$$

$$\sigma > \frac{4.605}{l_{T_s}}. \quad (3.19)$$

- peak time:

$$T_p = \frac{\pi}{\omega_d} < l_{T_s}, \quad (3.20)$$

$$\omega_d > \frac{\pi}{l_{T_p}}. \quad (3.21)$$

- maximum percent overshoot:

$$Mp = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} < l_{M_p}, \quad (3.22)$$

$$\zeta > \sqrt{\frac{\ln^2(l_{M_p})}{\pi^2 + \ln^2(l_{M_p})}}. \quad (3.23)$$

where l_{T_s} , l_{T_p} and l_{M_p} are the limits respectively. Considering the provided limits, some guidelines for seeking suitable poles can be obtained which probably result desired system dynamics. Rearranging the equations 3.13 and 3.14, the controller gain estimations can be calculated:

$$K_p = \frac{2\sigma\tau - 1}{A}, \quad (3.24)$$

$$K_i = \frac{\tau(\sigma^2 + \omega_d^2)}{A}. \quad (3.25)$$

As it was mentioned before, choosing the poles based on the above limits can not guarantee acceptable system behaviour as our system is not a canonical second-order, on the top of that the plan model is also simplified to a first order and not even any nonlinearity is considered.

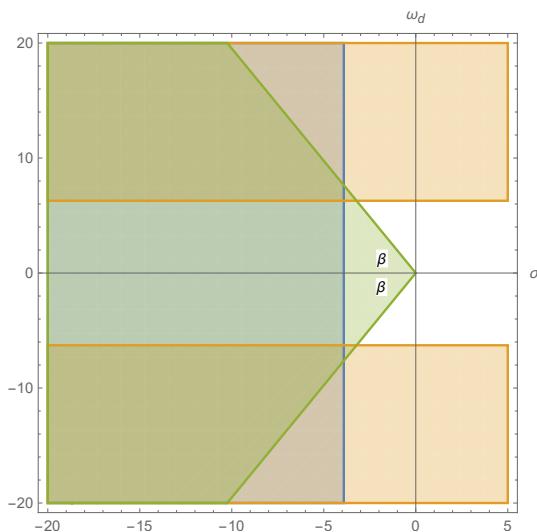


Figure 3.14: Pole parameter field considering the limits

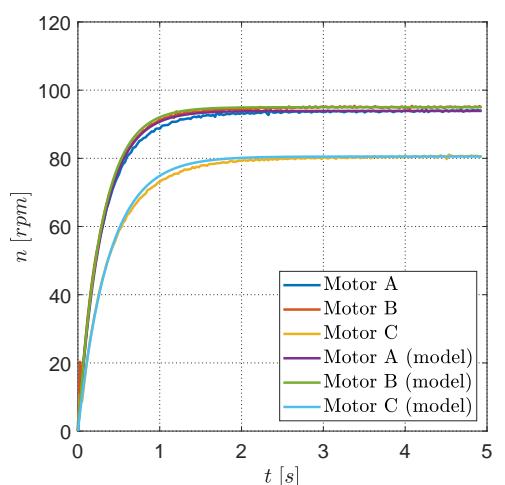


Figure 3.15: Black box model at 0.5 PWM value

The Figure 3.14 shows the possible real and imaginary value pairs which satisfies the requirements. The real part of pole is σ , the imaginary part is ω_d and the angle β can be calculated by means of $\zeta = \cos(\beta)$.

Let us have the following requirements:

- Settling time for 1%: 2[s]
- Peak time: 1[s]
- Max peak amplitude: 10%

and let us use measurements corresponding to pwm value of 0.5. After fitting the first order black box model the following values were obtained. Considering the given requirements,

	A	B	C
τ [s]	0.2949	0.2880	0.3767
k [rpm]	93.8978	94.9998	80.5325

Table 3.2: First-order model parameter estimations

as a first approximation placing the poles of the system to $\sigma = 5$ and $\omega_d = \pi$ point can be good. In this case, the $K_p = 0.021$ and $K_i = 0.11$ are obtained. The system responses

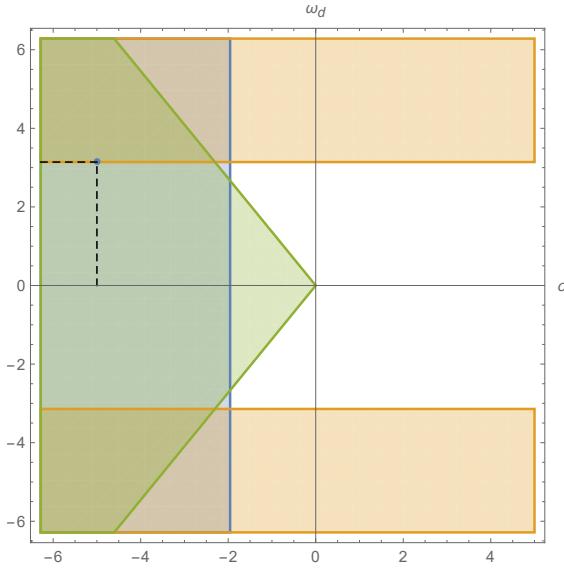


Figure 3.16: Possible poles that suits the requirements

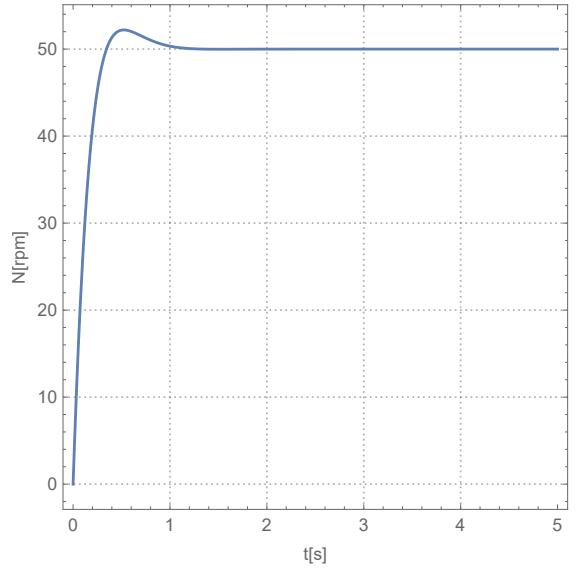


Figure 3.17: Expected response

are shown in Figure 3.18 and 3.19. In case of target speed of 100 rpm, the controller makes a really bad performance with so high oscillations. The oscillations vanishes if the integral type controller gain is set to zero, but the final value will be uncertain. After some trial and error for seeking a better integral gain an acceptable combination were found. The corresponding system responses are presented in Figure 3.20 and 3.21. The only preliminary criteria which was not fulfilled is the max peak amplitude in case of 50% PWM, when it was about 15% instead of 10%. However, better peak and settling times were obtained.

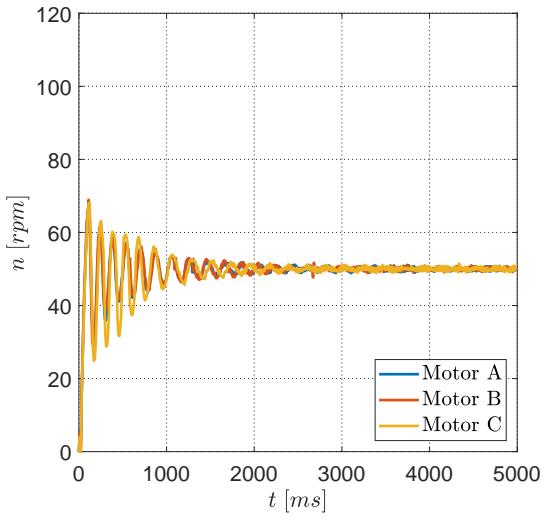


Figure 3.18: Obtained response at target velocity 50 [rpm]

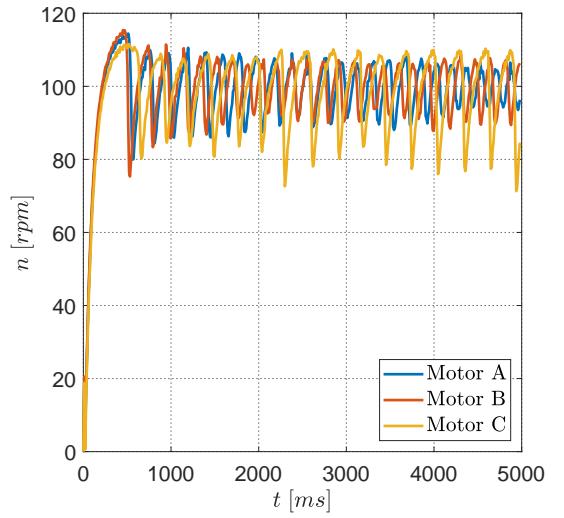


Figure 3.19: Obtained response at target velocity 100 [rpm]

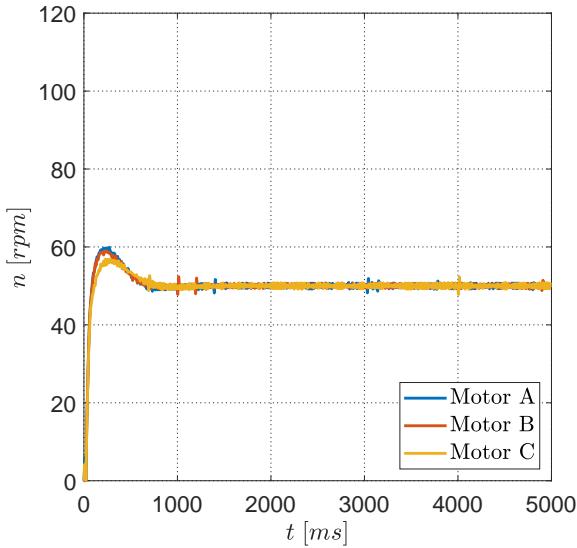


Figure 3.20: Obtained response at target velocity 50 [rpm]

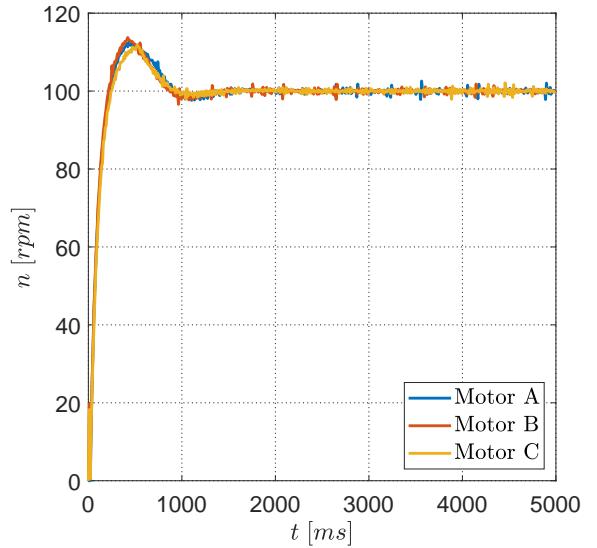


Figure 3.21: Obtained response at target velocity 100 [rpm]

4 Motion Analysis of The Robot

4.1 Kinematics of three-wheeled omnidirectional robot

4.1.1 Basic relationships

Let us define the global frame as $[X, Y]$ coordinate system which represents the environment. The location and the orientation of the robot can be described with (X, Y, α) or (X, Y, φ) vector, while the kinematic state is that of time-derivative, i.e. $(\dot{X}, \dot{Y}, \dot{\alpha})$ or $(\dot{X}, \dot{Y}, \dot{\varphi})$. Eventually, we are usually interested in the kinematics from global frame point of view, but from the design and user point of view it is more convenient to start writing the equations in the local frame of the robot and then transform the kinematics to the global frame if that is needed. Let us denote the local frame as (x, y) which center is the center of gravity of the robot. The robot velocity as v when that of components in the local frame are $(v_x, v_y) = (-v\sin\gamma, v\cos\gamma)$.

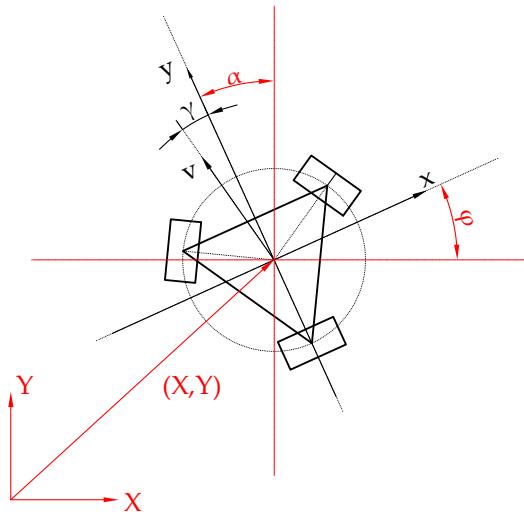


Figure 4.1: Global and local frames of the robot

The overall kinematic state of the case is a straightforward resultant of the wheel kinematic states. The velocity of wheels can have two type of components:

- Translational velocity which actually equals with translational speed of the case because of the kinematic constraint. Additionally, it can be divided into two further parts:
 - Its own velocity which can be called as pure translational wheel speed. It is proportional with the angular velocity of the wheel: $v_{wheel,t} \sim r\omega_i$, where r is the wheel radius and ω is the angular velocity.
 - The rolling velocity v_{roller} which is induced by the other wheels and realized by the free rollers. Its direction is determined by the construction of rollers, generally and also in this case let it assume to be 90° .
- Rotational velocity component which comes from the spinning of the whole robot case about its center of gravity which can be supposed to be in the geometrical center of the common wheel circle, therefore it can be formulated as $v_{wheel,r} = R\dot{\alpha}$. This component is always the same for all wheels.

The resultant translational wheel velocity can be calculated as:

$$v^i = v_{wheel,t}^i + v_{wheel,r} = r\omega_i \quad (4.1)$$

where i is the index of the wheels and $i=1,2,3$ in case of a three-wheeled omnidirectional robot.

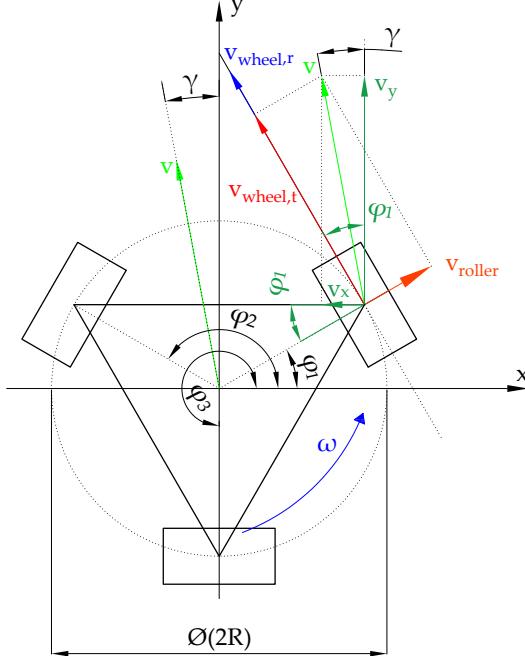


Figure 4.2: Wheel vectors in local (x,y) coordinate system

The omniwheels are located along the common wheel circle (R) at $\varphi_1, \varphi_2, \varphi_3$ angles measured from the local x axis. The pure translational part of the wheel velocity can be written as a combination of the x, y components of the robot translational velocity as:

$$v_{wheel,t}^i = -v_x \sin \varphi_i + v_y \cos \varphi_i. \quad (4.2)$$

Considering the equations above the total wheel velocity is:

$$v_i = -v_x \sin \varphi_i + v_y \cos \varphi_i + R\omega_i, \quad (4.3)$$

therefore the angular velocity of the wheel is:

$$\omega_i = \frac{1}{r}(-v_x \sin \varphi_i + v_y \cos \varphi_i + R\omega_i), \quad (4.4)$$

which can be generalized in vectorial form as:

$$\boldsymbol{\omega} = \mathbf{J}^{-1} \mathbf{v} \quad (4.5)$$

$$\mathbf{v}_i = r \mathbf{J}^{-1} \mathbf{v} = \mathbf{T}_{velocity} \mathbf{v} \quad (4.6)$$

where

- $\Omega = [\omega_1 \ \omega_2 \ \omega_3]^T$ is the wheel angular velocity vector,

- \mathbf{J}^{-1} is the inverse Jacobian matrix of the case velocities which is

$$\mathbf{J}^{-1} = \frac{1}{r} \begin{bmatrix} \sin\varphi_1 & \cos\varphi_1 & R \\ \sin\varphi_2 & \cos\varphi_2 & R \\ \sin\varphi_3 & \cos\varphi_3 & R \end{bmatrix}, \quad (4.7)$$

,

- $\mathbf{T}_{\text{velocity}}$ is the velocity transformation matrix,
- $\mathbf{v} = [v_x \ v_y \ \omega]^T = [\dot{x} \ \dot{y} \ \dot{\phi}]^T$ is the case velocity vector.

The inverse of determinant of \mathbf{J} is never zero, thus its inverse always exists which means that kinematic relation between the wheels and the case is provided. In order to track the robot in the global configuration, the kinematic state has to be transformed from the local frame to the global frame which can be done by means of a simple rotation transformation perpendicular to (x,y) plane with α orientation angle:

$$\mathbf{x} = \mathbf{T} \ \mathbf{X} \quad (4.8)$$

, i.e.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Phi} \end{bmatrix}. \quad (4.9)$$

4.1.2 2D Simulations

Considering the equations above, a two-dimensional kinematic simulator was built-in Wolfram Mathematica which could help to understand more accurately the robot locomotion mechanism by calculating and then visualizing the movements. The simulator has to be initialized with as follows:

- Common wheel circle (R) which is in this case: $\text{caseR} = 0.12 \text{ m}$,
- Wheel position angles (φ_i) which is in this case $\varphi_1 = 30^\circ, \varphi_2 = 150^\circ, \varphi_3 = 270^\circ$,
- Initial position and orientation $\mathbf{X} = [X \ Y \ \alpha]^T$, i.e. $\mathbf{p} = [X \ Y]^T$ and α ,
- Initial velocity state in the local frame: $\mathbf{v} = [v_x \ v_y \ \omega]^T$, i.e. v, γ and ω .
- The solution is obtained numerically, integrating the velocities over time steps, thus timestep size (dt) and simulation time (T) has to be provided.

The basic equations are the followings:

- The new position equation is

$$\mathbf{p}^{i+1} = \mathbf{p}^i + \mathbf{T}\mathbf{v}^i dt, \quad (4.10)$$

- The new orientation equation is

$$\alpha^{i+1} = \alpha^i + \omega^i dt. \quad (4.11)$$

- The new velocity magnitude equation is

$$v^{i+1} = F(v^i, dt, \mathbf{p}^i, \dots). \quad (4.12)$$

- The new velocity direction equation is

$$\gamma^{i+1} = F(\gamma^i, dt, \omega, \dots). \quad (4.13)$$

- The new angular velocity equation is

$$\omega^{i+1} = F(\omega^i, dt, \mathbf{p}^i, \dots). \quad (4.14)$$

The Figure 4.3 depict a pure translation in forward direction. An interesting characteristic of the robot configuration can be observed, i.e. the resultant case velocity can be greater than the wheel velocities. The Figure 4.4 represent the case of pure rotation when the wheel velocities are equal. The Figure 4.5 depicts another specific case of pure

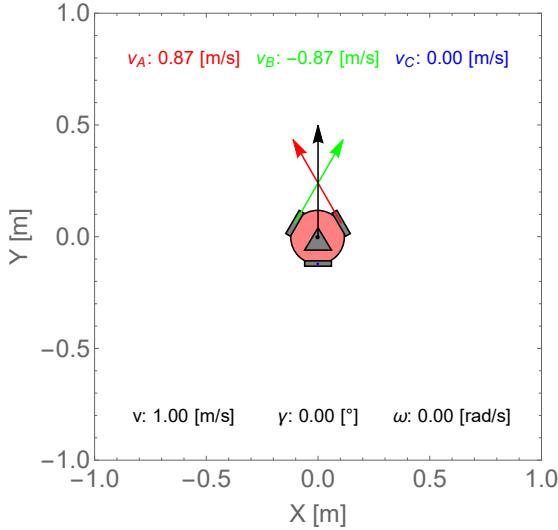


Figure 4.3: Pure forward translation

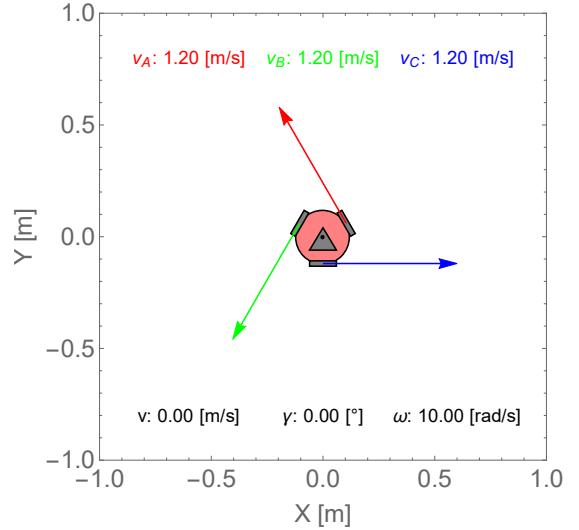


Figure 4.4: Pure spinning

translation when the velocity of wheel A equals the robot velocity, while the others drive with same speed but the opposite direction. The Figure 4.6 represent a mixed case of translation in an arbitrary direction and rotation at the same time.

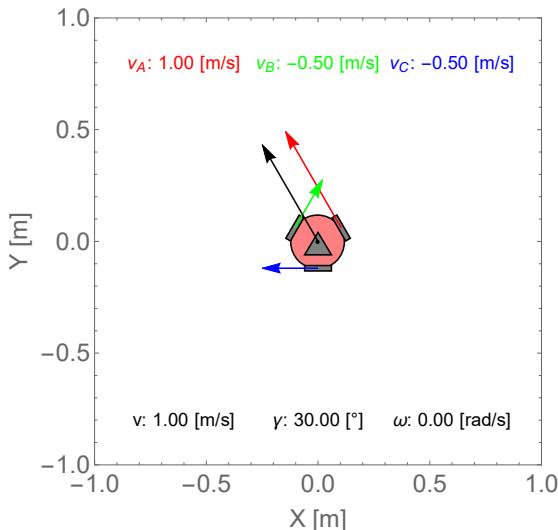


Figure 4.5: Specific case of translation

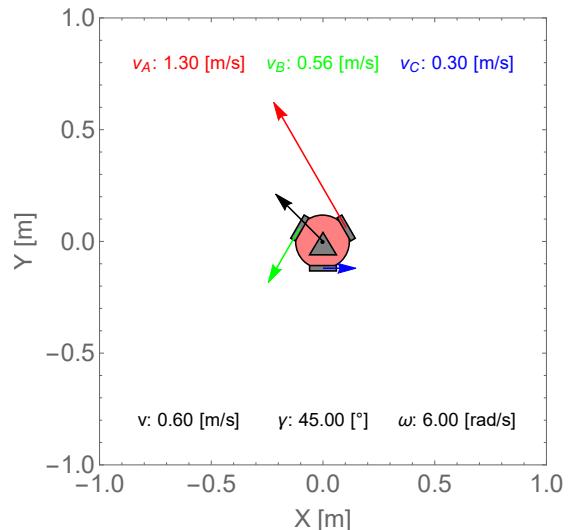


Figure 4.6: Translation & rotation

The following figures represent the status of robots at given time steps during the simulations. The main equations that generates the shown motions are given. Generally there are 3 parameters to vary: the magnitude of the speed, the direction of the speed and the angular velocity.

Moving along a circle without changing the orientation:

- $dt = 0.1$ [s] (timestep size), $T = 10$ [s] (simulation time),
- $v^{i+1} = v^i = v^0$ (constant speed),
- $\omega^{i+1} = \omega^i = 0$ (no angular velocity),
- $\gamma^{i+1} = \gamma^i + \dot{\gamma}dt$ where $\dot{\gamma} = \frac{2\pi}{T}$

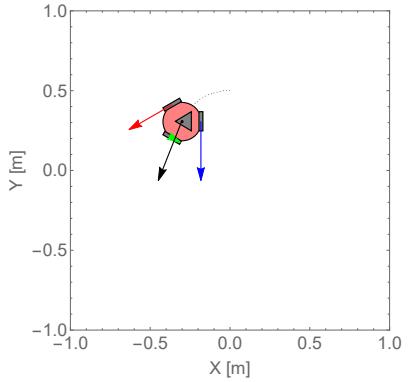


Figure 4.7: 0.20[s]

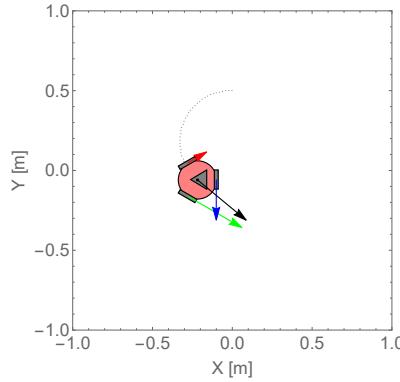


Figure 4.8: 0.40[s]

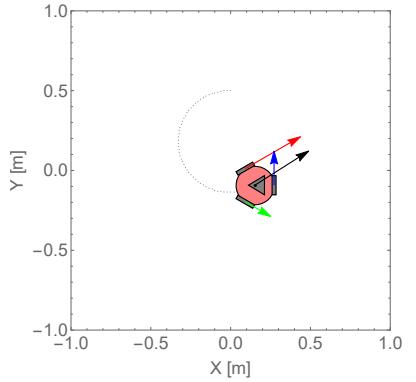


Figure 4.9: 0.60[s]

The direction of the speed vector (γ) is defined in the local coordinate system and it has to be changed to make the robot move along a given curve, in this case along a circle.

Moving along a circle without like a car (anholonomic):

- $dt = 0.1$ [s] (timestep size), $T = 10$ [s] (simulation time),
- $v^{i+1} = v^i = v^0$ (constant speed),
- $\omega^{i+1} = \omega^i = \omega^0$ (constant angular velocity),
- $\gamma^{i+1} = \gamma^i = 0$

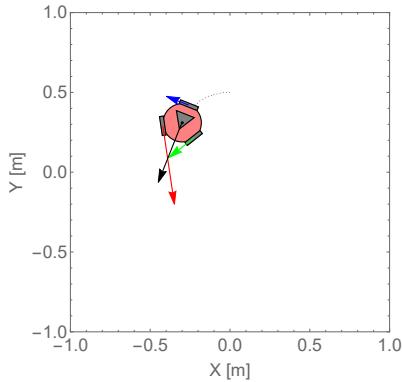


Figure 4.10: 0.20[s]

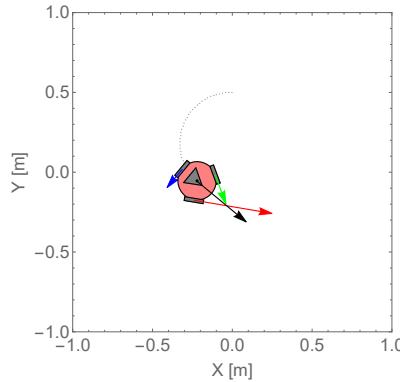


Figure 4.11: 0.40[s]

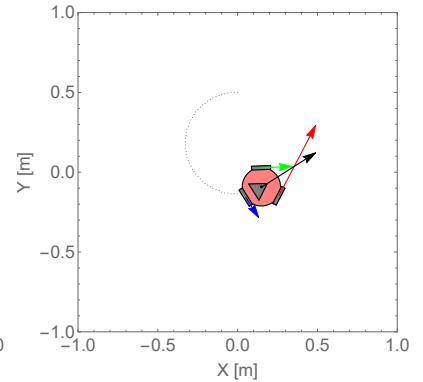


Figure 4.12: 0.60[s]

The following two cases are further demonstration of the fact that the orientation can be controlled independently from the the orientation, i.e. the robot has 3 DoF.

Moving along a circle changing the orientation in reverse direction:

- $v^{i+1} = v^i = v^0$ (constant speed),
- $\omega^{i+1} = \omega^i < 0$ (angular velocity),
- $\gamma^{i+1} = \gamma^i + \dot{\gamma}dt - \omega^i dt$ where $\dot{\gamma}$ is a constant.

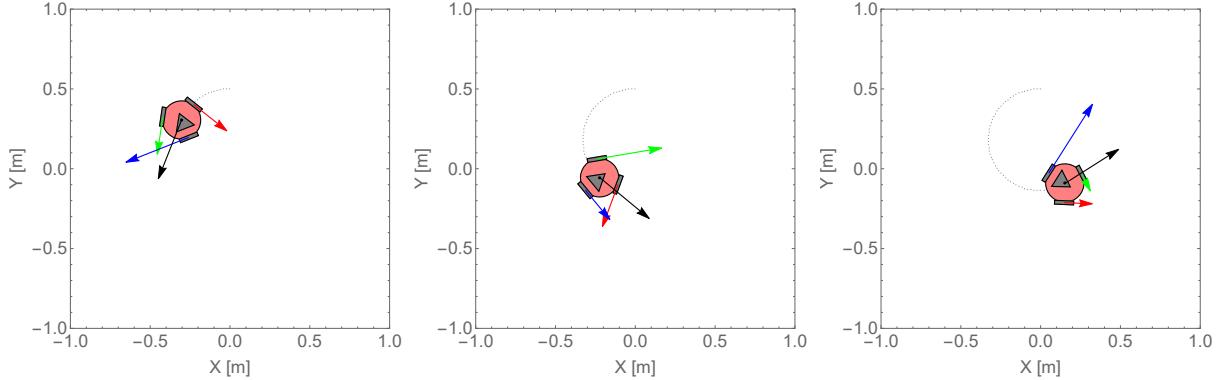


Figure 4.13: 0.20[s]

Figure 4.14: 0.40[s]

Figure 4.15: 0.60[s]

In this case also the robot orientation and also the speed direction is changing from step to step. The changed orientation (ωdt) has to be considered in the equation of γ as being defined in the local frame.

Moving along a line and changing the orientation:

- $v^{i+1} = v^i = v^0$ (constant speed),
- $\omega^{i+1} = \omega^i < 0$ (angular velocity),
- $\gamma^{i+1} = \gamma^i - \omega^i dt$ where $\dot{\gamma}$ is a constant.

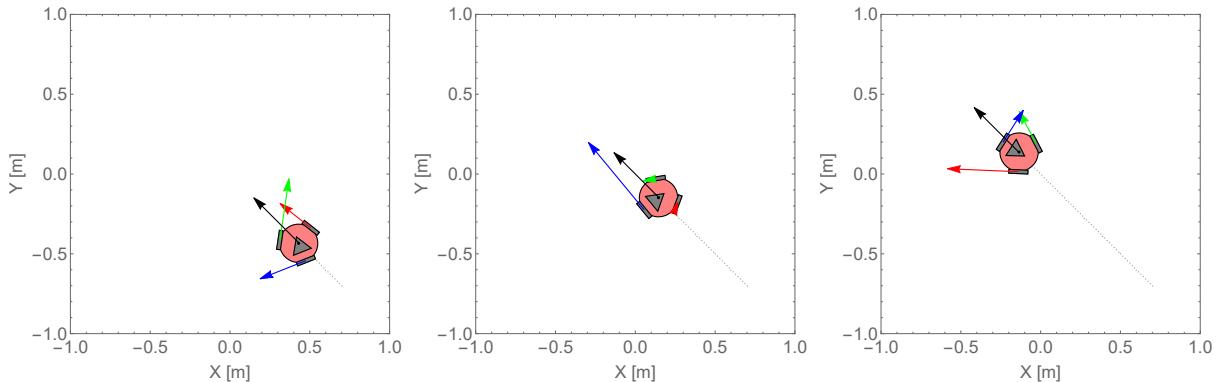


Figure 4.16: 0.20[s]

Figure 4.17: 0.40[s]

Figure 4.18: 0.60[s]

The kinematic simulation code is a handy tool to demonstrate the possibilities of such a robot configuration, but it does not give any details from dynamics point of view. As it can be seen in previous chapter the real world is not as ideal, thus such a simple kinematic point of view is not enough to give a detailed analysis.

4.2 Dynamic analysis of DC motors

In order to have a proper dynamic model, the source of forces, i.e. the motors have to be analysed and modeled in details. As it was mentioned before, the permanent magnet type DC motors are easy to be described using 5 parameters (L, J, R, B, K), thus the primary goal is estimate their values.

4.2.1 DC motor parameter identification

The L (inductance) and J (second moment of inertia) describes the dynamic behavior of the motor, while the others characterize the system in a stationary aspect. Based on motor datasheet informations (no load case, max efficiency case, max power case, stall torque case), only the stationary parameters can be estimated and you can even not be sure about the values whether they match the reality or not. Obviously, to get know about the dynamic parameters, measurements of dynamic operations of the motors are needed, e.g. measuring the step response to a given function of voltage over time. A robust technique for the parameter identification is to combine the measurement results with the capabilities of MATLAB Simulink and its Design Optimization tool. To carry out the following models and the parameter identification process the following references were considered: [22], [23], [24]. The Simscape library of the Simulink makes it possible to make system models of coupled physical problems combining submodel elements, e.g. a DC motor which is an electromechanical system can be built using simple elementary mechanical and electrical components.

Firstly, in accordance with the equations of DC motor a basic model was made. The

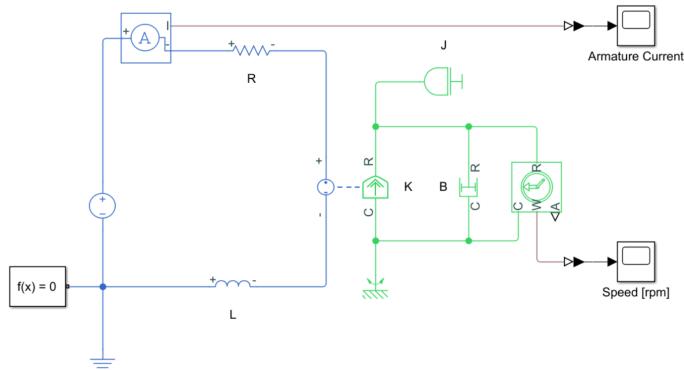


Figure 4.19: Simple DC motor electromechanical model in Simulink

Figure 4.19 represent the first model in which the blue is the electrical circuit consisting a resistance (R), inductance (L), DC voltage source, electrical reference and current sensor, while the green elements are mechanical blocks, i.e. the inertia (J), rotational damper (B), rotational reference and a rotational speed sensor. The interface between the two domains is provided by the electromechanical converter which defines the K constant. The $f(x)$ block defines solver settings for the numerical solution and the other two elements are blocks that monitors the current and shaft speed over time. This model has to match with the governing equations because actually those are solved numerically. The armature current has high peaks at the starting of DC motors due to the absence of the back emf which is proportional with rotational speed. The high gradients make the equation system to be a stiff one, thus `ode15s` solution method was chosen which is a recommended one for such problems by means of the MATLAB help.

In order to use the parameter estimation tool of the software, more complex voltage inputs have to be defined. This model works with simple constant DC input voltage only.

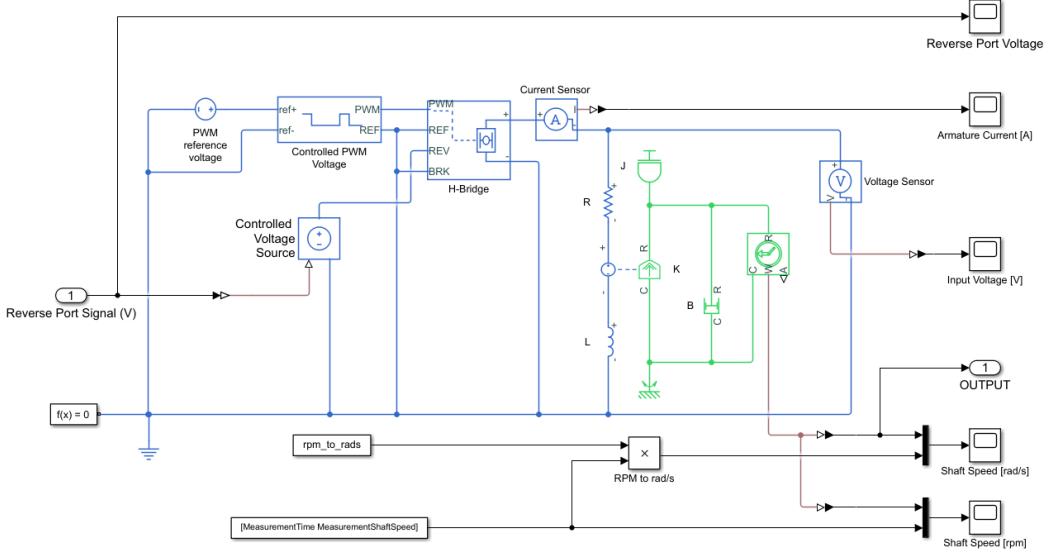


Figure 4.20: Complex DC motor electromechanical model in Simulink

From the Simscape library, further electrical components can be added to the model such as the H-Bridge which can switch the rotational direction of the rotor or such as the PWM module which can produce a square wave with a given duty cycle to control the magnitude of the input voltage. The model shown in the Figure 4.20 has two inputs. One is the PWM reference signal which is a voltage value between 0 and 5V. The Controlled PWM unit generates from the reference signal a square wave which will be the input of the H-bridge motor drive component. This part can be related to a voltage value of PWM type digital pin of the MCU. The following component, the H-bridge motor drive, is actually the model of the L298N driver. The magnitude of its output voltage is controlled by the PWM signal's duty cycle, while that of polarity is controlled by the digital signal at REV port. The second input of the model is the Reverse Port Signal which has to be a digital signal, thus 0V (LOW) correspond to no polarity change and 5V (HIGH) corresponds to switched polarity, thus reverse rotation of the rotor. The output of the model is shaft speed considered in rad/s. Transient operation of the system is provided by changing the reverse port digital signal.

In order to make the parameter estimation with the specified inputs, measurements for the output speed had to be taken before the simulation could be run. In addition, the parameters has to be initialized in the model as a preliminary guess to have a combination of them to start the optimization from somewhere. Based the governing equations and motor data sheet values, a good initial guess could be given. In case of no load operation, $b = \frac{T}{\dot{\theta}}$ where $T = 0.11$ [Nm] and $\dot{\theta} = 11.21$ [rad/s] (107RPM), therefore $b = 0.01$ [$\frac{\text{Nms}}{\text{rad}}$]. Also other working points are known with respect to T-i values pairs, thus supposing linear relation the K can be estimated which was calculated to be 0.9338. Using these two values and supposing the steady state version of the Kirchoff law in case of no load, the resistance can be calculated as $R = \frac{V - K\dot{\theta}}{i} = 10.25$ [Ω]. Considering all above, the initial guess of the sought parameters is as follows: $R = 10$ [Ω], $b = 0.01$ [$\frac{\text{Nms}}{\text{rad}}$], $J = 0.001$ [kgm^2], $L = 0.01$ [H], $K = 1$. In the parameter estimation module both of the measurement and the simulation of the system can be represented which is shown in the Figure 4.21. The plots on left side represent the simulation with the initial parameters. It can be seen that the steady-state value is close to the measured one, but the time-scales are quite different. The plot on the right side correspond to the estimated values which are the followings:

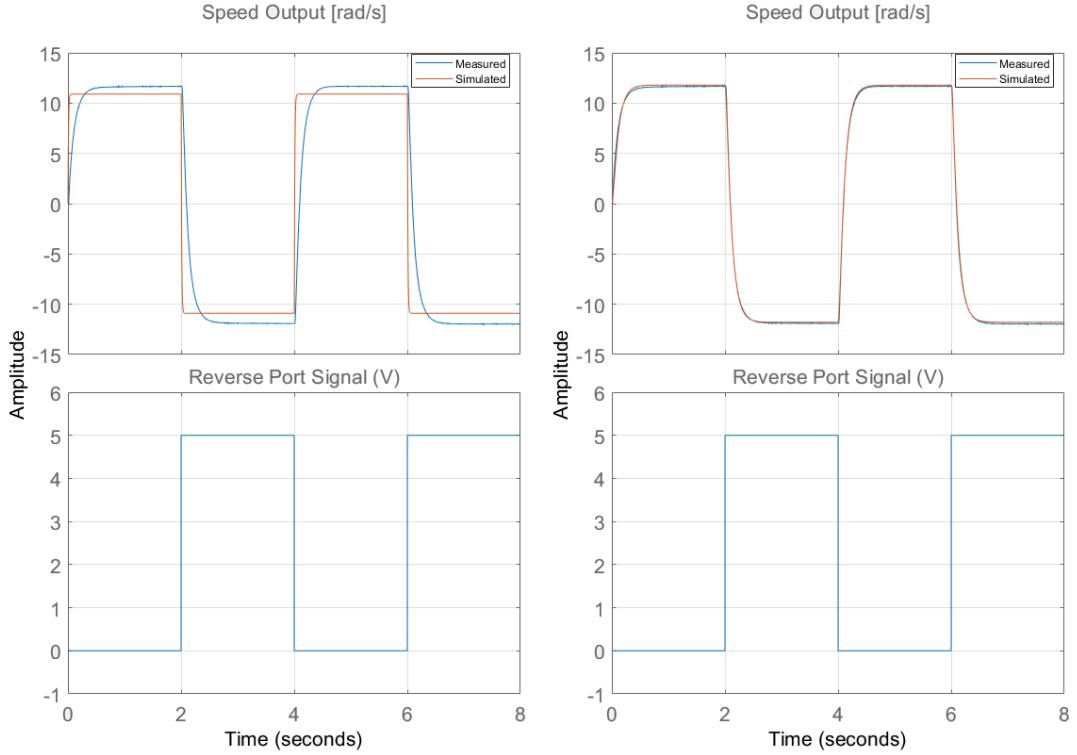


Figure 4.21: Simulation results before and after the optimization

$R = 7.98[\Omega]$, $b = 0.0197[\frac{\text{Nms}}{\text{rad}}]$, $J = 0.0121[\text{kgm}^2]$, $L = 0.166[H]$, $K = 0.825[\frac{\text{Vs}}{\text{rad}}]$. The sum of squared error was selected to be the cost function which converged from the initial value of 66.6 to 0.23 after 11 iterations. The obtained values are in a good relation with estimates, thus with the possible real values. The steady state current in the simulated free load case is 2 times higher, but the peak current is in a good order. Really high peaks can be got when the polarity of the input voltage is switched, therefore higher torques can be expected during such operations which means more risk for fall over for instance. The initial peak current can be estimated as $U/R = 12[V]/10.25[\Omega] = 1.17[A]$ assuming zero initial shaft speed. Figure 4.23 shows a similar current peaks in the simulation.

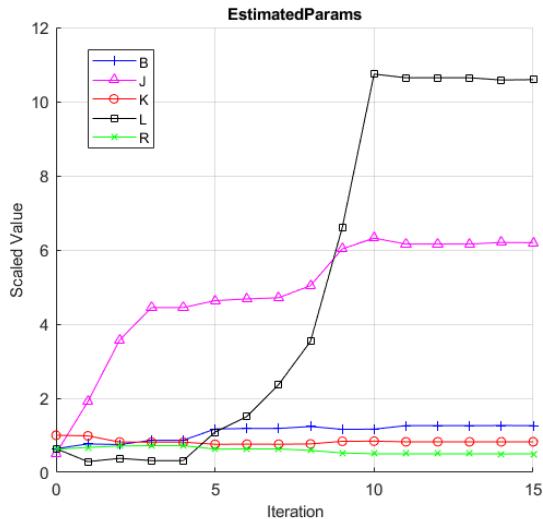


Figure 4.22: Iterations of estimated parameters

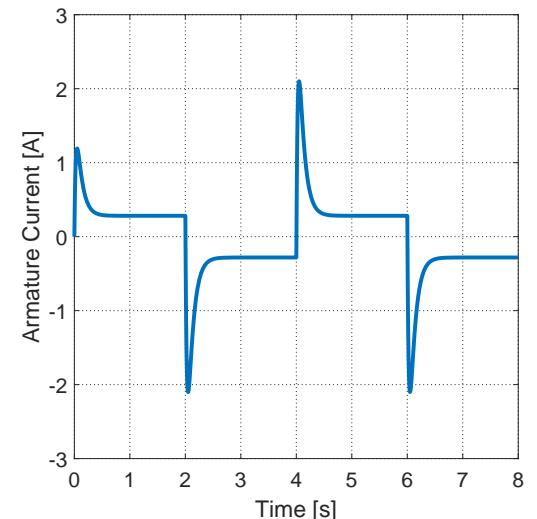


Figure 4.23: Armature current of the estimated model

The solution method was got numerically while the dynamic model of the robot can be solved analytically in special cases. Generally a more simple method is seeked that might be implemented also in the hardware code, thus it would be more convenient to find a good analytical formula too calculate the current, thus the motor torques exerted on the wheels. Firstly, in a similar way the numerical solution of the governing equations were implemented in Wolfram Mathematica to have a more flexible control on the solution and to have the possibility to implement it in a mechanical model later on. Because of the high current peaks, implicit Euler-method was chosen combined with Newton-Raphson iterations because the built-in solver could not handle stiffness of the equation properly. The chosen method calculated the same dynamics for the system as the numerical solution in Simulink. The dynamic system model is as follows:

$$\begin{bmatrix} \dot{\omega} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} -\frac{b}{\theta} & \frac{k}{\theta} \\ -\frac{k}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} + \begin{bmatrix} -\frac{T_{LOAD}}{\theta} \\ \frac{V}{L} \end{bmatrix} = \begin{bmatrix} f_1(t, \omega, i) \\ f_2(t, \omega, i) \end{bmatrix} = \mathbf{f}(t, \omega, i). \quad (4.15)$$

Let us denote $\mathbf{x} = [\omega, i]$. Applying the discretization by means of Implicit-Euler technique the following equations are obtained:

$$\begin{bmatrix} F_1(t, \mathbf{x}_k, \mathbf{x}_{k+1}) \\ F_2(t, \mathbf{x}_k, \mathbf{x}_{k+1}) \end{bmatrix} = \mathbf{x}_{k+1} - \mathbf{x}_k - h\mathbf{f}(t_{k+1}, \mathbf{x}_{k+1}) = \mathbf{F}(t_{k+1}, \mathbf{x}_k, \mathbf{x}_{k+1}) \quad (4.16)$$

where

- \mathbf{x}_k is the current time step variable vector,
- \mathbf{x}_{k+1} is the next time step variable vector,
- h is the time step,
- t_{k+1} is time at the $(k + 1)^{th}$ timestep

Using the Newton's method an inner iteration equation of a given step is as follows:

$$\mathbf{x}_{k+1}^* = \mathbf{x}_{k+1} - \mathbf{J}_{\omega, i}^{-1}(\mathbf{x}_{k+1}) \mathbf{F}(t_{k+1}, \mathbf{x}_k, \mathbf{x}_{k+1}) \quad (4.17)$$

where

$$\mathbf{J}_{\omega, i} = \text{grad } \mathbf{F}(t_{k+1}, \mathbf{x}_k, \mathbf{x}_{k+1}). \quad (4.18)$$

The solution can be calculated by means of ?? by proceeding in time and at the given time steps the solution of the algebraic equation is obtained by Newton's iteration using the equation ???. The h time step was chosen to be 0.001s and the 3 inner iteration were used in equation ??? which provided sufficiently accurate solution. Considering the governing equations, it can be said that let us omit the inductance of the armature, thus the "inertia" of the electric circuit. With this assumption combining the two constitutive equations, the following first-order linear differential equation for $\omega[t]$ rotational speed can be derived:

$$b\omega(t) + \theta\dot{\omega}(t) = \frac{K(V - K\omega(t))}{R}, \quad (4.19)$$

which solution is:

$$\omega(t) = \frac{e^{-\frac{bt}{\theta} - \frac{K^2 t}{\theta R}} \left(KV e^{\frac{t(bR + K^2)}{\theta R}} + bR\omega_0 + K^2\omega_0 - KV \right)}{bR + K^2} \quad (4.20)$$

where ω_0 is the initial angular speed. The armature current can be extracted as

$$i_a(t) = \frac{V - K\omega}{R}. \quad (4.21)$$

The Figure 4.24 and 4.25 depict the numerical and analytical solution for the shaft speed [rad/s] and armature current [A] where the orange solid line is the analytical and dashed black is the numerical solution.

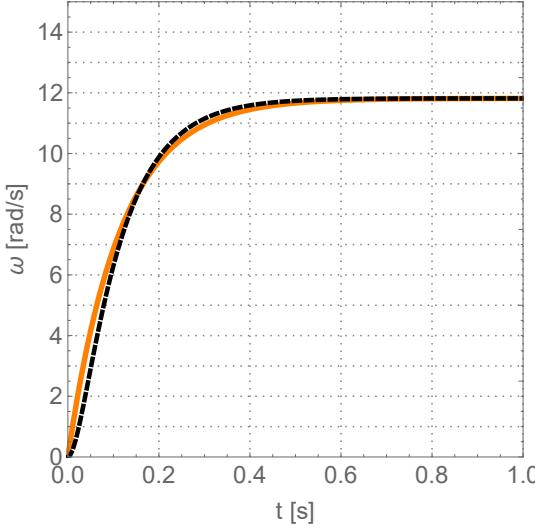


Figure 4.24: Angular velocity

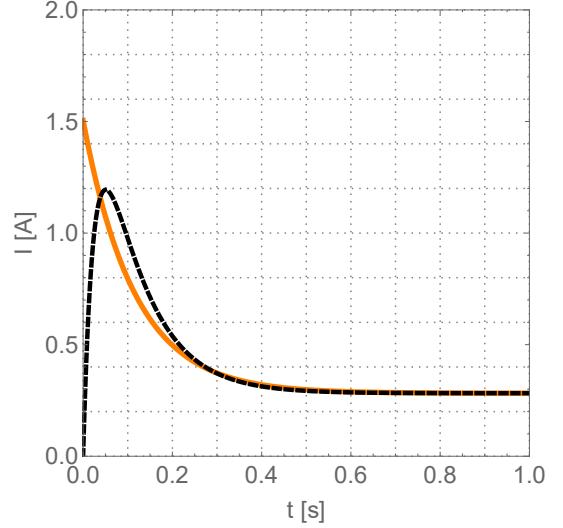


Figure 4.25: Armature current

The analytical solution is in an acceptable match with the more accurate numerical one (where L is not assumed to be zero) and also higher current is estimated at start which is a conservative engineering approach. In this way, it is a good estimation to work with the analytical formula in the dynamic model of the robot. The following general relationship between torque and input voltage can be derived:

$$T(t, U, \omega_0, T_{Load}) = \frac{K \left(-\frac{Ke^{-\frac{t(bR+K^2)}{\theta R}} (\omega_0(bR+K^2) - KU + RT_{Load})}{R} + bU + KT_{Load} \right)}{bR + K^2} \quad (4.22)$$

where

- t [s] is the time,
- U [V] is the voltage,
- ω_0 [rad/s] is the initial shaft speed,
- T_{LOAD} [rad/s] is the load torque,

4.2.2 Validation of the parameter estimation

4.2.3 Rolling resistance measurements

4.3 Mechanical model of the robot

A simplified dynamic model was made which is shown in Figure 4.26. Its main purpose to provide a tool that can evaluate the given states of the robot and estimate the risk for the fall over. The robot has a set of fixed properties, thus its inertias are known. As being a transport robot, it has a given payload which has to be taken to a destination without losing the stability. The package attached to the robot will change the mechanical properties and it can happen that it falls over if too high acceleration are caused. On one hand, this model is not aimed to describe the details of the fall over of the robot. In order to handle that process, a timed advanced solution should be provided by integrating the equations over time where due the change of the orientation around the global x and y axis the problem would became nonlinear which solution would be complicated and time-demanding. The mechanical model should be much more detailed to capture properly the changing inertias of robot, moreover the location of contact points would not be trivial to determine too. All in all, it would be too complicated and make no sense as it is not even wanted that the robot could work at the edge of the stability. From practical point of view, it is enough to search for an indicator that correlates well with the risk of fall over. For instance, monitoring the sign of the calculated normal force (N_i) between the wheels and the ground is logical choice. If its sign changes to negative from positive then the robot is possible at edge of the fall over. The negative sign has no real physical meaning since the ground contact is such a geometrical constrain that can only push but not pull. On the other hand, it is a convenient engineering approach to think as if it could work like that because then the closure of the system of equations is provided easily and simple to be solved. This kind of concept is reasonable since the more significant part of the configuration will be the package and that of the effect on the common center of mass will dominate.??? TODO!

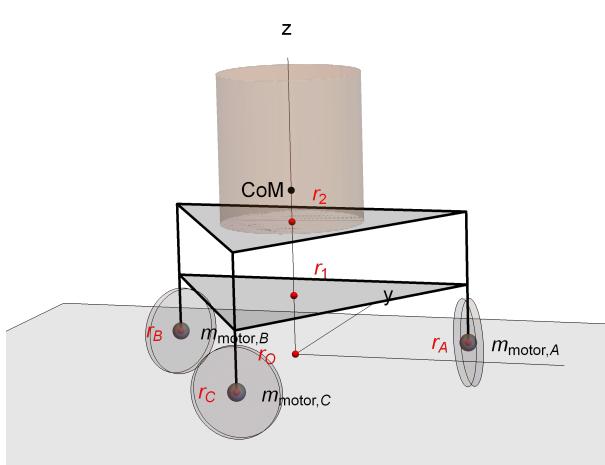


Figure 4.26: Simplified mechanical model of the robot

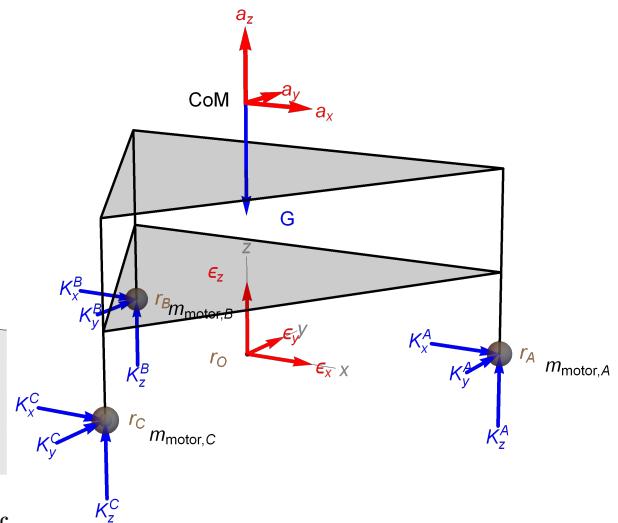


Figure 4.27: The case free body diagram

4.3.1 Model of the robot case

The robot model consist of the case, that of free-body diagram (FBD) is represented in Figure 4.26 and three wheels which are shown in Figure 4.28. The wheels are connected to the case via simply joints which can transfer only force components, i.e. no moments. For instance, the pair of the motor torque exerted on the motor housing due to Newton's 3rd Law, thus on the bracket which is connected to the case with screws is also not considered

hereby since its magnitude is about 1% because of the gearbox which has a ratio of 1/90. (TODO: or to take it into consideration) The origin (r_O) of the global frame is defined as a center of gravity of the wheel center points (r_A, r_B, r_C) when the orientation angle of the robot is zero, as in the Figure 4.26 it is represented.

- The levels of the robot configuration are considered as equilateral triangles. The second moment of inertia around the z axis of a single triangle at its own mass center can be calculated as $\theta_z = \frac{mR^2}{3}$ where $R = 0.12$ [m] is the common wheel radius. The other components of the inertia matrix is calculated as if it was a disk for simplicity. As it was mentioned, the fall over is not calculated, thus these components are ignored in the calculation, only the z component is important.
- The motors and that of brackets are considered as point mass at the wheel center points.
- The package is considered as a (set of) cylinder shape object having $\theta_z = \frac{m_p r_p^2}{2}$. The contact between the case and the package is bonded.

The equations of Newton's 2nd Law at common center of mass for the robot case is as follows:

$$m_R \mathbf{a} = \mathbf{K}_A + \mathbf{K}_B + \mathbf{K}_C + \mathbf{G} \quad (4.23)$$

where

- m_R [kg] is the common mass of the case and the package,
- \mathbf{a} [$\frac{\text{m}}{\text{s}^2}$] is the acceleration vector,
- \mathbf{K}_i [N] is the force transferred at wheel connections, $i=(A,B,C)$,
- \mathbf{G} [N] is the weight force, $G = gm_C$.

$$\theta_R \varepsilon + \omega \times (\omega \times \theta_R \omega) = (\mathbf{r}_A - \mathbf{r}_{\text{CoM}}) \times \mathbf{K}_A + (\mathbf{r}_B - \mathbf{r}_{\text{CoM}}) \times \mathbf{K}_B + (\mathbf{r}_C - \mathbf{r}_{\text{CoM}}) \times \mathbf{K}_C \quad (4.24)$$

where

- θ_R [kg m^2] is the common second moment of inertia matrix at the center of mass (CoM), $\theta_R = \theta_{\text{Case}} + \theta_{\text{Package}} + \theta_{\text{Motor A}} + \theta_{\text{Motor B}} + \theta_{\text{Motor C}}$ (each inertia matrix of a given body is calculated at its own center of mass and then transformed to the common center of gravity),
- ε [$\frac{1}{\text{s}^2}$] is the angular acceleration,
- ω [$\frac{1}{\text{s}}$] is the angular velocity,
- $\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C$ are the wheel position vectors and \mathbf{r}_{CoM} is the position vector of the common center of mass which is

$$\mathbf{r}_{\text{CoM}} = \frac{m_{\text{Triangle}}(\mathbf{r}_1 + \mathbf{r}_2) + 3 \times m_{\text{Motor}} \mathbf{r}_O + m_{\text{Others}} \frac{(\mathbf{r}_1 + \mathbf{r}_2)}{2} + m_{\text{Package}} \mathbf{r}_{\text{Package, CoM}}}{m_{\text{Case}} + m_{\text{Package}}} \quad (4.25)$$

where

- $m_{\text{Case}} = 2 \times m_{\text{Triangle}} + 3 \times m_{\text{Motor}} + m_{\text{Others}}$,
- $m_{\text{Triangle}} = 0.184$ [kg] is the mass of the triangle-shaped case elements,
- $m_{\text{Motor}} = 0.200$ [kg] is the mass of the motor and the bracket,

- $m_{Others} = 0.4$ [kg] is the mass of the other electronic hardware elements which are considered as if it was in the center of mass of the triangles,
- $m_{Package}$ is the arbitrary mass of the package.

There are 15 unknowns while 6 equations are known up to this point:

$$a_x, a_y, a_z, \varepsilon_a, \varepsilon_b, \varepsilon_c, K_x^A, K_y^A, K_z^A, K_x^B, K_y^B, K_z^B, K_x^C, K_y^C, K_z^C.$$

The mechanical equations are considered for each wheels in their local coordinate system which definition is depicted in the Figure 4.28. Applying the Newton's 3rd Law and taking

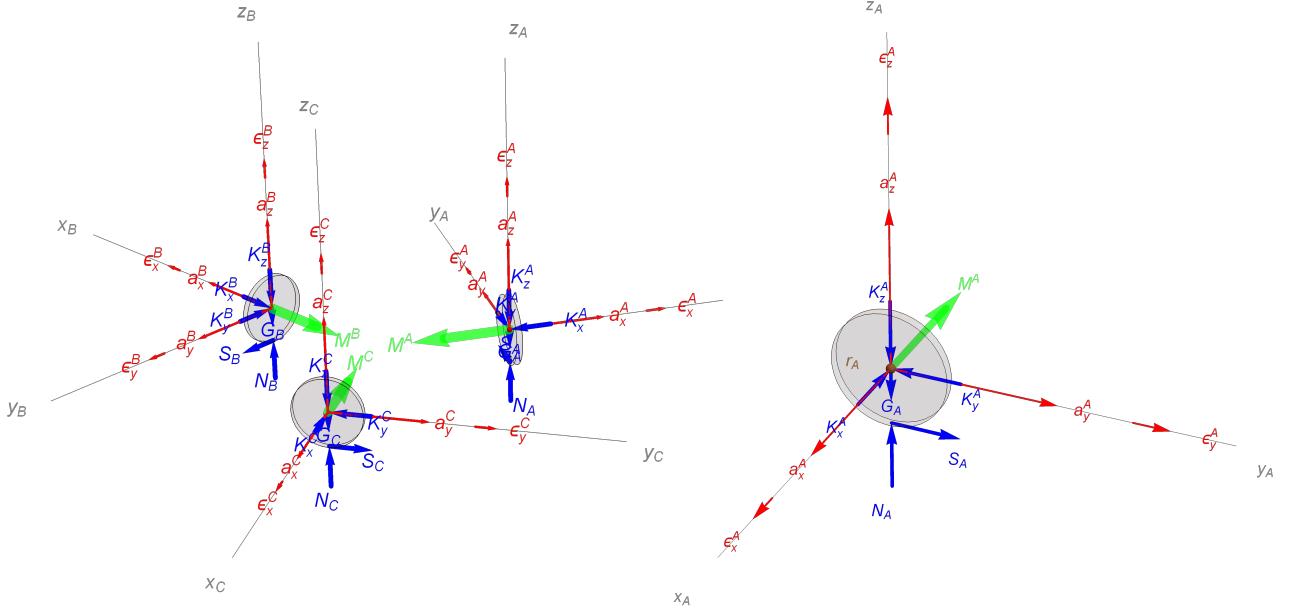


Figure 4.28: The free-body diagrams of the wheels

Figure 4.29: The free-body diagram of the wheel A

into consideration the given coordinate system transformations, the following relationship can be written regarding the transferred forces in the joints:

$$\mathbf{k}_i = - \begin{bmatrix} \cos(-\varphi_i) & -\sin(-\varphi_i) & 0 \\ \sin(-\varphi_i) & \cos(-\varphi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{K}_i = \boldsymbol{\sigma}_i \mathbf{K}_i \quad (4.26)$$

where \mathbf{k}_i is the force in the joint in the local wheel frames and $i = [A, B, C]$. $\boldsymbol{\sigma}_i$ is the rotational transformation matrix between the global and local frame. In other words, the components of \mathbf{k}_i will not be new unknowns having the relationship above. Beside the forces in the connection, the gravitational body force and the contact forces (N, S) are considered in case of the wheels with assumed y, z directions respectively. (TODO:literatura) As a rough approximation, simple one-point contact and no slip condition is assumed which provides six kinematic constrain equations:

$$a_y^i = -r\varepsilon_y^i \quad (4.27)$$

$$a_z^i = 0 \quad (4.28)$$

where $i = [A, B, C]$.

In the axial direction the moving friction force is not considered. In reality in case of an omni wheel, there are rollers in a circumferential alignment which can also spin. In this

model, they are modeled as simple frictionless sliders. The possibility of the slippage of wheels are neglected as it would need precise measurements to have a dynamic friction coefficient.(TODO:literatura) On the other hand, the so called rolling resistance can be defined as an adverse torque from the motor load torques (M^i) point of view which could be taken into consideration in the DC motor equations as loads.

The Newton equations for the wheels ($i = [A, B, C]$) are the followings:

$$m_i \mathbf{a}_i = \mathbf{k}_i + \mathbf{S}_i + \mathbf{G}_i + \mathbf{N}_i \quad (4.29)$$

where

- \mathbf{k}_i [N] is the force transferred by the joint,
- \mathbf{S}_i [N] is the static friction force,
- \mathbf{G}_i [N] is the gravitational force,
- \mathbf{N}_i [N] is contact force.
- $m_i = 60$ [g] is the mass of a single wheel,
- \mathbf{a}_i [$\frac{\text{m}}{\text{s}^2}$] is the acceleration,

$$\theta_i \varepsilon_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \theta_i \boldsymbol{\omega}_i) = -\mathbf{M}_i + \mathbf{r}_{\mathbf{S}_i} \times \mathbf{S}_i \quad (4.30)$$

where

- $\boldsymbol{\theta}_i$ [kg m^2] is the second moment of inertia matrix of the wheel at the center of mass (CoM), its axial (x) component come from the DC parameter identification, i.e. the total inertia of the rotating part is considered. The other components are calculated as if it was a cylinder with the known geometry and mass. This simplification is acceptable because of the fact that the fall over is not wanted to be treated by the model.
- ε_i [$\frac{1}{\text{s}^2}$] is the angular acceleration of the wheel,
- $\boldsymbol{\omega}_i$ [$\frac{1}{\text{s}}$] is the angular velocity of the wheel,
- \mathbf{M}_i is the motor torque exerted on the rotating part of the motor and the wheel.
- \mathbf{S}_i is the static friction force at the contact of the wheel and ground, while $\mathbf{r}_{\mathbf{S}_i}$ is the vector pointing to the contact point from the wheel center of mass (\mathbf{r}_i).

There are 24 additional unknowns which are the followings:

$$a_x^i, a_y^i, a_z^i, \varepsilon_x^i, \varepsilon_y^i, \varepsilon_z^i, S^i, N^i \text{ where } i = [A, B, C].$$

The Newton's 2nd Law provides 24 equations (6 for each wheels), the rolling constrains means 6 equations, thus in order to have the closure of the system 6 more equations are needed. The wheel-case joint can be assumed to be a rigid body connection in which case the following equations stand and make the closure:

$$\mathbf{a}_i = \boldsymbol{\sigma}_i(\mathbf{a} + \boldsymbol{\varepsilon} \times \mathbf{r}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_i)) \quad (4.31)$$

The motor loads can be calculated by means of the Eq. 4.22, thus the coupling can be done leading to an electrodynamic model. The analytic approach is used as that is an overestimation due to the higher angular acceleration peaks, faster as only the evaluation

of an analytic formula is needed and it is possible to calculate only a time-moments without having to know the preceding history of motion. Eventually, the inputs of the model are inertias of component (mass, second moment of inertia), motor velocity states and the voltages at the pins of the motors, meanwhile the output is the acceleration state of the whole robot. The desired voltages can be determined by means of the velocity transformation matrix derived in the previous chapter (see Eq. 4.6) which was about the kinematics. This can be generalized since the wheel speed is proportional with the applied voltage, therefore a prescribed motion can be carried out by applying the voltages in the same ratio as in an idealistic model the velocities should be applied.

4.3.2 Simulations of motor states

The Figure 4.30 represents a solution of a simple forward acceleration case from rest. It can be seen that the contact forces of wheel A,B decreases while that of wheel C increases. This phenomenon can be explained by the inertial forces which tries to rotate the case and at the same time the increased contact forces withstand in order to prevent the rotation. In case of too high acceleration compared to the height and the inertias of robot, this phenomenon could lead to fall over. If the magnitude of acceleration is kept constant, the corresponding distribution of contact forces will depend on the direction of the acceleration vector. The Figure 4.31 depicts the parametric solution of the contact forces with respect to the γ angle of the acceleration vector. The dependence can be described with trigonometric functions and it can be concluded that the most dangerous cases, i.e. minimum values of contact forces are present when the robot accelerates such a way that the direction of the acceleration vector is coincident with a single wheel direction. It is possible that by changing only orientation of the robot, the model equations calculates negative contact forces. As it was mentioned, negative forces has no real physical sense, they are obtained because of the kinematic constrains but they can be regarded as an indicator for a possible risk for fall over. The ground constrain works as if it was able to exert pulling force to prevent the rising of the case. It can be an unexpected problem when the robot tries to move along a curved path and the so called centrifugal acceleration have the robot lost its stability. A possible safety feature can be for such a robot to change its own rotation along the curve curve that it takes into consideration the direction of the current acceleration and tries to set such an orientation where the reduction of contact forces are equally distributed between two wheels instead of one as in the worst case. The angular acceleration does not affect the contact forces based on this model, thus such movements for correction purposes can be done safely and independently.

The Figure 4.30 depicts the case with the most favorable orientation with respect to the evolution of the contact forces. Despite the fact that this model could predict a risk for fall over even in this configuration, in the reality this can hardly happen. In this case the wheel A and B could rise above the ground due to the too high acceleration but as they are actually the ones that causes the accelerations also the cause would vanish. Moreover, the wheel C is just perpendicular to the direction of the acceleration, thus its passive rollers are aligned such a way that they can roll freely. The direction of the rolling just after the rising of the wheels, thus the rotation of the robot depends on the location of the common center of mass. The Figure 4.32 shows the two possible cases. On one hand, the gravitational force is always balanced by the contact force on the passive wheel. On the other hand, the resultant moment, thus the direction of the angular acceleration depends on the orientation of the geometry, i.e. the relative position center of mass. If the robot case does not rotate that much that the center of mass does not get beyond the passive wheel than the robot will fall back and roll towards the passive wheel. Therefore,

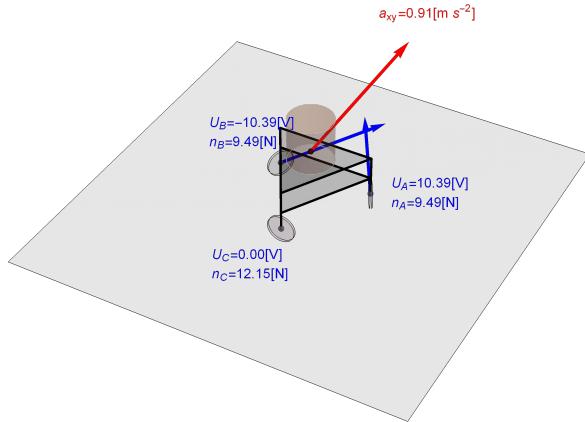


Figure 4.30: State at forward acceleration

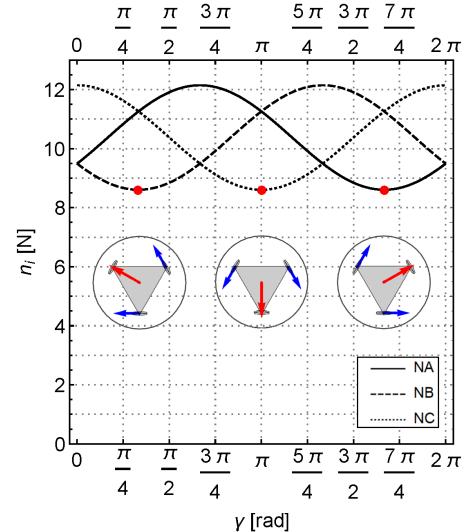


Figure 4.31: The free-body diagram of the wheel A

the opposite situation can not even happen by an inertial force which is caused by a too high acceleration as the passive will roll in the opposite direction immediately after the contact of wheel A and B are lost. Although, in reality there are mostly such conditions when this prediction based on this given idealistic model does not hold properly. The case is not perfectly rigid, thus it will vibrate and it may deform such a way due to a sudden inertial force that the center of mass can get beyond the wheel. In addition, the passive wheel can also stick to a given surface if it is not perfectly aligned or the passive roller is not works as an idealistic roller, i.e. as a slider from global point of view if the moving friction is not considered.

In any kind of other situation, the robot can theoretically fall over even in this configuration if it gets a specific amount of angular momentum by an external disturbance. In that case, the resultant of the obtained angular momentum and the effect of the gravitational torque will determine whether it falls over or not. This problem is highly nonlinear due to the change of orientation and needs a time-advancing solution to have analysis about the possible outcome, which is here not the goal to deal with.

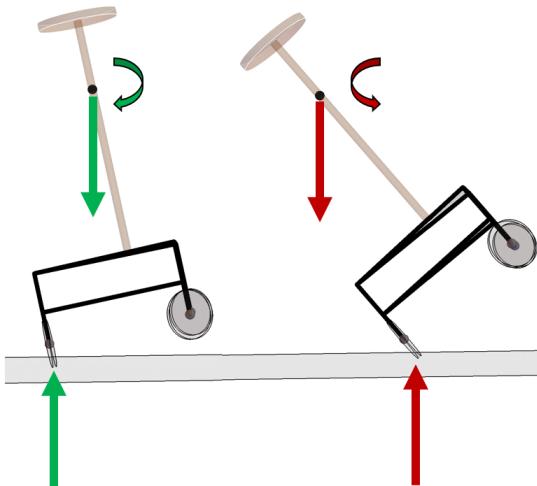


Figure 4.32: Dependence of the location of center of mass

There can be cases when the orientation of the robot cannot be changed or simply not changed for any reason and the acceleration of the robot is oriented in the most disadvantageous direction. It is disadvantageous as this is the direction in which the robot can tilt more easily and due to the alignment of the driving wheels, the robot can tilt itself by exerting high torques on the driving wheels as the driving wheels keep their contact with the ground during the fall over as well. Using the derived model an autonomous robot can predict for a given input if too high accelerations are expectable and control the motors to avoid the risky manoeuvres. Let us consider that the given robot has a disk shaped mass on a rod which is bonded to the case just as in the Figure 4.32. This construction was motivated by only demonstrative reasons as it could be realized in the hardware as well using a threaded rod and disk shaped weights. If the resultant center of mass is located at a higher vertical level, the motor will more likely to fall over for the same accelerating torques. In order to raise the center of mass point, added extra mass above the robot center of mass is needed. Needed means in this case, the need for the analysis to have a configuration that generally operates at the edge of its stability in certain events and then it tries operate such a way to avoid the possible risks. A possible case can be when the robot moves with a constant speed using two driving wheels just in the risky orientation then suddenly the driven wheels are braked by applying a voltage step from a given constant value to zero. The robot starts decelerating and the inertial force tries to lean it over. By the way, this is more or less equivalent case to a constant acceleration from rest in the opposite direction.

The Figure 4.33 shows that in case of max braking power, the model predicts negative contact forces for the passive wheel with the given inertias. Applying a smaller voltage step to a non zero initial value then decreasing that of value exponentially provides a smooth and not risky acceleration. The simulation in case of negative contact forces are strictly invalid, they are just a simplified tool for prediction for fall over. If no negative contact forces in the simulation are captured, then the obtained characteristic can be regarded as an estimation having similar trend of the real ones, i.e. physical quantities change by exponential trend if idealistic conditions are assumed. In reality, the wheel speed reaches the constant zero value due to the presence of static friction. It is important to emphasize

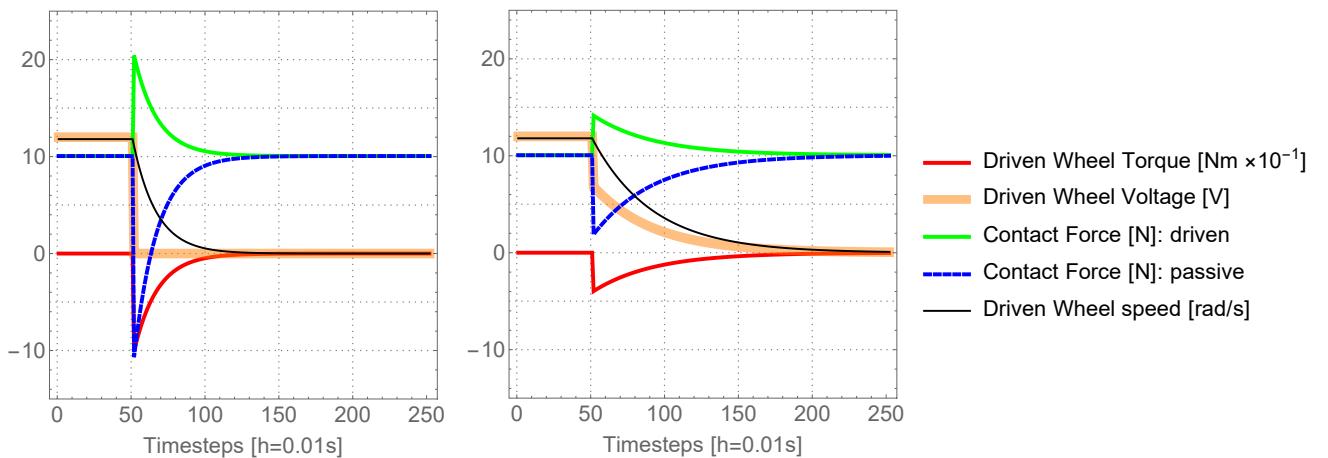


Figure 4.33: Braking with max torque and reduced torque in case of 1.0[kg] disk at the top of the rod at 1[m] height

that the presence of the negative forces does not guarantee that the robot will fall over. If the negative forces last for only moments, in other words the passive wheels is not in contact with the ground anymore, there will not be enough work from driven wheels to overcome the reversing torque exerted by the gravitational force. Eventually, it can be

concluded that all in all a conservative engineering approach to tune the braking profile such a way that the no negative forces are calculated by the model. The tuned voltage profile in the simulation is obtained such a way that resultant torque profile is the one-third of the maximal one at each time step, thus automatic and smooth decrement is also set. If mass of the disk or that of position along the rod or the length of the rod changes, this calibration will likely not work. If only the mass of the disk is the changing variable in the system, then combining the above force prediction model with a mass estimator one, a more general prediction tool can be got.

4.3.3 Comparision with measurements

In order to test the derived dynamic model whether it is in a good relation with the reality or not, measurement were held using the built robot configuration. The robot started to move from rest and reached its maximum speed during 2 [s] then stopped applying a step to zero voltage with a steep falling edge resulting in a high torque on wheels, thus high acceleration at the center of gravity of the whole configuration. In order to shift the center of gravity in a higher position, a threaded rod was applied with an additional disk-shaped masses attached to the end. The order of the center of mass displacement can be controlled by applying higher mass at the top. The configuration with the rod that has 0.373 [kg] mass and is attached by a disk that has 0.5 [kg] mass, was proven to be able to resist the braking condition which is shown in Figure 4.34. The passive wheels rose above the ground slightly but the robot's center of gravity did



Figure 4.34: Braking with max torque and reduced torque in case of 0.5[kg] disk at the top of the rod at 1[m] height

not rotated beyond the passive wheel, therefore it returned to the ground after some oscillatory bouncing. From practical point of view, this behavior is also undesired since it is a risky way of stopping and such movements can cause damages. Then, the mass at the top was doubled and the robot fell over without any douts which shown in Figure 4.35. The dynamic model provided an initial peak of passive wheel contact force of -8.05[N] for 1[kg] and -2.88[N] for 0.5[kg]. The derived dynamic model predicted negative forces, thus passive wheel rising which was experienced during the measurements. It is also favorable that the same tendency was captured which also a good verification of the model. From engineering point of view, the model provides estimation of fall over risk in a conservative manner if avoidance of negative signs of contact forces is the main design

challenge. With the given weights also the fall over was tested in case of opposite direction

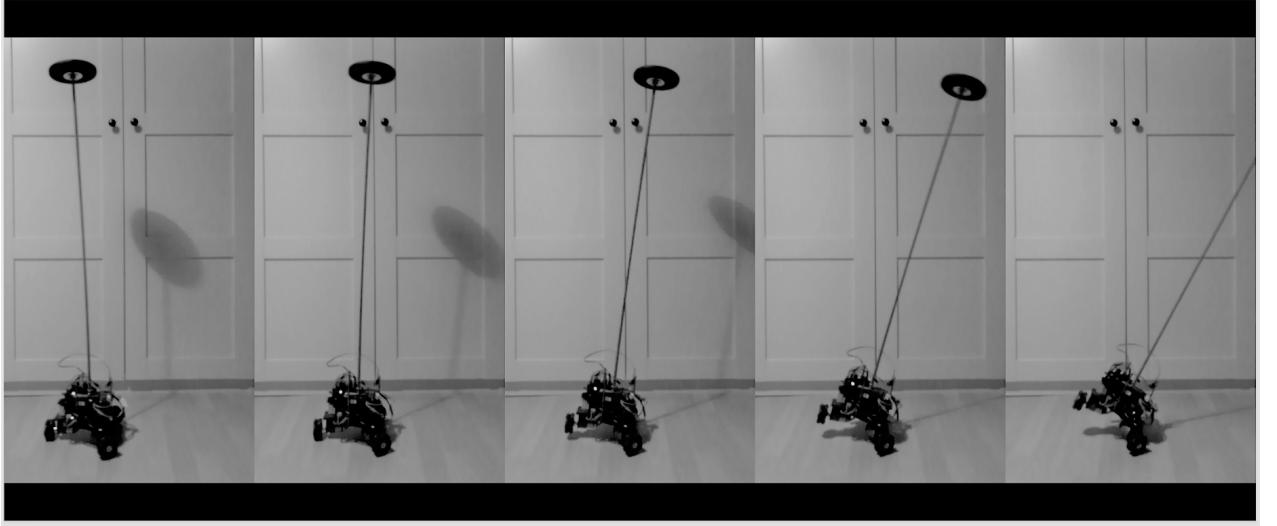


Figure 4.35: Braking with max torque and reduced torque in case of 1.0[kg] disk at the top of the rod at 1[m] height

acceleration, i.e. in such configuration when the contact force distribution is theoretically the most favorable. As it was mentioned, the simplified rigid and one-point contact robot model automatically predicts it impossible to happen. With the given weights it was unable to fall over. Just after the braking, the two driven wheels rose above the ground for a little time. Due to the intense vibration of the rod, the robot changed a random amount of its orientation angle in clockwise or counter-clockwise direction by rotating around the third passive wheel's contact point(s). On one hand, it was verificated by the measurement that the robot can fall over in a specific direction and it cannot do like that in the corresponding opposite direction if the same inertias and accelerating power is used. On the other hand, due to the undesired random rotation around the passive wheel, the robot changes its orientation a little bit which is likely not acceptable in a real-life situations. Consequently, working at the edge of stability in this case is also a state which is wanted to be avoided during operation since it can lead to inaccurate orientation control and unpredicted movements. Generally, the contact force depends on the mass of the attached disk and that of height of attachment in case of a given robot configuration and the acceleration state. The Figure 4.36 represents the contact force of the passive wheel during the braking case as a function of the disk mass and height of attachment. It can be seen that by increasing either of them causes a reduction of contact force, thus increased risk for fall over.

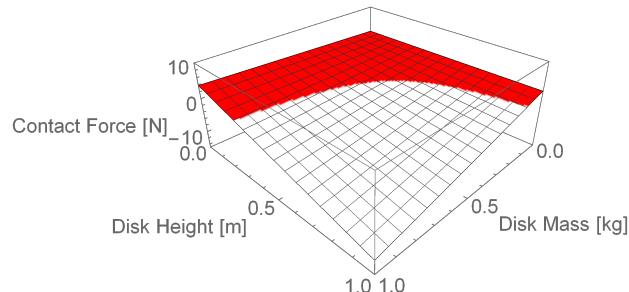


Figure 4.36: Passive wheel contact force (red means positive)

In this demonstrative configuration, it can be said that the height of the attachment

point of the disk is kept constant, then only the disk mass remains as an unknown if the acceleration state is known.

4.3.4 Mass estimation

A more general model can be developed if a mass estimation algorithm is provided. For instance, the acceleration of the robot case has to depend on the robot mass. Although, the dependence is not trivial as the common acceleration is the resultant of three accelerating wheels. Parametric solution for the initial acceleration in case of the braking is shown in Figure 4.37. Compared to the weight of the robot (which is about 2[kg]), a quite high mass of payload is needed to have noticeable difference in the initial acceleration peak. In this consideration the rolling friction is not included, the acceleration dependence is caused by changes of inner joint force components only. In my case, the robot hardware was able to have extra load about 2[kg] without having too high case deformations that can be harmful, thus this configuration is probably not able to work in a range where remarkable derivatives of the acceleration state could be experienced. Another problem is that in the model a simplified geometrical approach is used, while in the reality the peak is not at the very first moment. Additionally, the height of the peak is hard task to measure accurately using the rotary encoder since it uses double differentiation. The Figure 4.38 represent measurement and simulation results of a no load case for a motor.

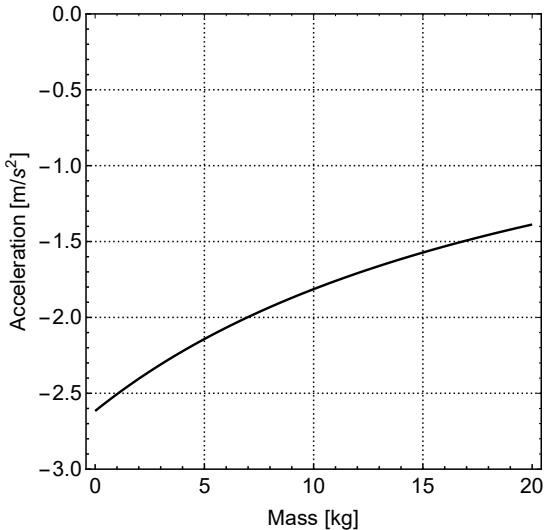


Figure 4.37: Mass of payload dependence of acceleration state

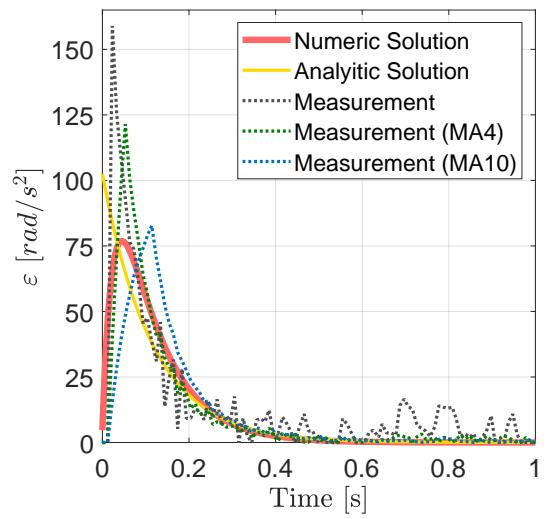


Figure 4.38: Comparision of angular accelerations of the motors

It is enough to analyze a single wheel dynamics as the resultant linear acceleration of the case is a that of resultant. The pure measurement is quite noisy as such high accelerations are calculated even if the the velocity of the wheel is constant. A moving average filter was tested to make correction on the data. If the averaging is taken on the two previous and proceeding timesteps (MA4) then the oscillation in case of steady state vanishes and the peak gets closer to the simulated ones. On the other hand, also the peak time is shifted which originally matches quite well with the numerical simulation. The same tendency can be experienced with a longer averaging function (MA10). Then the same measurements were carried out with package loads up to 1.7 [kg]. As it was mentioned, for structural reasons it was the maximum that this simple mobile robot could hold without harmful deformations. The measurements results showed little difference regarding the wheel accelerations. As in the Figure 4.37 shown, the relative changes of the initial

peak of linear acceleration as a function of payload of the given robot hardware is too small compared to the measurement and processing errors (numerical differentiation) of the rotary encoders. It can be stated that the errors are more significant than the sensitivity to the payload mass, therefore monitoring the acceleration state is not suitable for developing a mass estimator algorithm.

Itt még elszünteti a rolling resistance mérést... - végülis biztonságosabb egy helyben mozogni - hasonlóan kicsit érzékeny... - egy adott felületre be kell kalibrálni - a mért érték talán jobban megbízható ha sokáig átlagolok

Measurements... TODO

Summary

- Kerekek szabályozásának implementálása
- Felboruáls állapotának elkerülése
- fix kapcsolatos kialakítás, modell alapján gyorsulások számítása
 - *** egyszerűbb modell, erről van egy kezdetleges
 - *** veszélyes tartományban üzemel a robot, mivel már borulnia kell, hogy mérjünk valamit
 - *** legegyszerűbb megoldás, hogy a tömeget megmérjük lineáris gyorsulásból, a geometria ismert
- rugós felfüggészetes kialatkítás, modell alapján max gyorsulások számítása
 - *** nem veszélyes tartományban mér a robot
 - *** bonyolultabb modell, rugalmas alapot is belekell rakni, adott gerjesztése adott válasz vizsgálata, rezgéses modell, kapott válasz alapján m, θ becslése
- borulás elkerülés pusztán mérések alapján,
self-balancing üzemmód, nincs benne mechanikai egyenlet
- mechanikai modell alapján optimális orientáció számítása (kész) és annak változtatása menet közben adott pálya mentén (laterális gyorsulás), motor nyomatékokból ki lehet számolni a pillanatnyi gyorsulást elvileg – ez is inkább szabályozási feladat már
- esetleg valami teljesítmény optimalizálós dolog

References

- [1] Dr. Tevesz Gábor: *Robotirányítás rendszertechnikája*
- [2] Roland Siegwart, Illah R. Nourbakhsh: *Introduction to Autonomous Mobile Robots*
- [3] James M. Anderson, Nidhi Kalra, Karlyn D. Stanley, Paul Sorensen, Constantine Samaras, Oluwatobi A. Oluwatola: *Autonomous Vehicle Technology*
- [4] *Autonomous car*: https://en.wikipedia.org/wiki/Autonomous_car
- [5] <http://iiot-world.com/artificial-intelligence/five-challenges-in-designing-a-fully-autonomous-system-for-driverless-cars/>
- [6] SMP Robotics: <https://smprobotics.com/> [link]
- [7] Starship Technologies: <https://www.starship.xyz/> [link]
- [8] NEOBOTIX: <https://http://www.neobotix-robots.com/> [link]
- [9] BMW STR: <http://www.bmwblog.com/> [link]
- [10] Allonrobots: <http://www.allonrobots.com/military-transportation-robots.html> [link]
- [11] Alan G. Smith: *Introduction to Arduino, 2011*
- [12] Brian W. Evans: *Arduino Programming Notebook, 2007*
- [13] Alex Lange: *I2C Communication with Arduino, 2015*
- [14] <http://howtomechatronics.com> [link]
- [15] Kevin Townsend: *Adafruit BNO055 Absolute Orientation Sensor, 2017*
- [16] <https://www.elprocus.com/what-are-the-best-ways-to-control-the-speed-of-dc-motor/> [link]
- [17] <https://nathotron.wordpress.com/courses/engineering-design-process-530-381/motor-selection/> [link]
- [18] <http://ctms.engin.umich.edu/CTMS/index.php> [link]
- [19] Dynamic Model of a Permanent Magnet DC Motor: <http://users.isr.ist.utl.pt/~alex/micd0506/motordc.pdf> [link]
- [20] Phidgets: DC Motor And Controller Primer
- [21] <https://www.mathworks.com/> [link]
- [22] <https://www.mathworks.com/videos/estimating-parameters-of-a-dc-motor-68856.html>
- [23] <https://www.mathworks.com/videos/modeling-a-dc-motor-68852.html>
- [24] <https://www.mathworks.com/products/simscape.html>
- [25] https://www.control.isy.liu.se/student/tsiu61/file/pm_dcmotor.pdf

[26] Magnus Jonason Bjärenstam: Development of a ball balancing robot with omni wheels, 2012 Lund University

[27] Arduino Documentation: <https://www.arduino.cc>