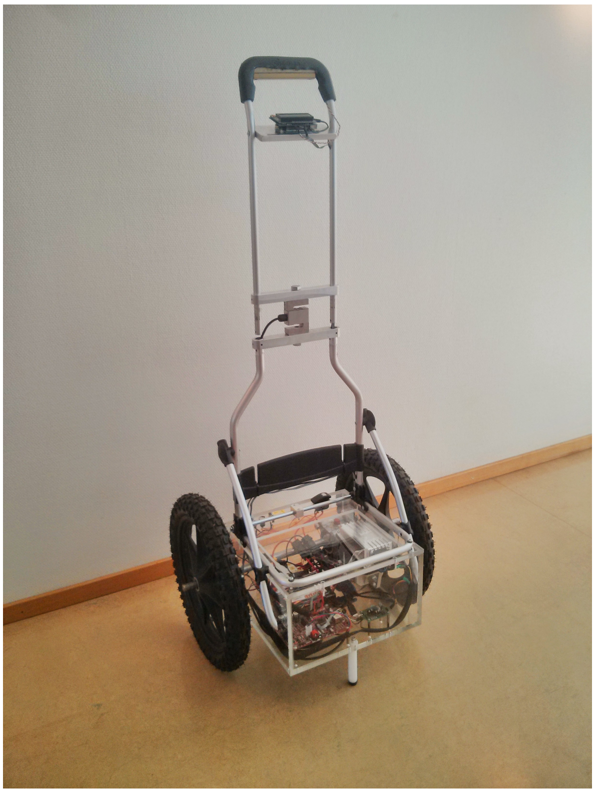
CHALMERS



Shopping Cart Dynamic Assistance System

Master's Thesis in Systems, Control and Mechatronics

JAKOB FJELLSTRÖM & ADAM MAANINKA

Department of Automatic Control, Automation & Mechatronics
Mechatronic Systems & Control Design
Chalmers University of Technology
Gothenburg, Sweden 2014

EX023/2014

Abstract

A shopping trolley with a dynamic assistance system is developed to help the elderly. The system gives help when the shopping trolley is heavy loaded and hard to pull and also helps when climbing stairs. The product doesn't give enough help in stairs due to low maximum motor torque or high trolley weight. A future development of the trolley might be to use two motors instead of one which is used in this prototype.

A load cell is implemented in the bar leading up to the handle in order to measure the force applied by the user. The measured force is then translated into a suitable helping torque provided by an electric motor. The stair climbing feature is achieved by using big wheels with knobby tires (rough texture) to grip the edges, no other mechanical solution is needed in this case.

Three different types of controllers are investigated in this report, proportional velocity control, integral velocity control and torque control. The conclusion is that the proportional velocity and torque controllers work well when climbing stairs and steep hills with the trolley. The integral velocity controller however, does not work well in stairs but is great when it comes to flat surfaces and hills with some inclination. The overall performance of the trolley is good and it fulfils its purpose.

Acknowledgements

We would like to thank Jonas Fredriksson who was our supervisor at Chalmers University of Technology, for giving us support and feedback. We would also like to thank Andreas Pettersson, Daniel Chädström and Johan Åström for being our supervisors at Aros Electronics and for their help and support during the master thesis.

We would like to thank Balázs Kulcsar for his thoughts during the development of the controllers, Birger Johansson, Roland and Tommy for the help during the mechanical design and construction, Per Lundberg, Martin Ahlqvist, Anders Bonde and Thorbjörn Simonsson for the help during the electrical construction and also David Hallberg, Jesper Fors, Oscar Leon, Sebastian Holmqvist and Conny Dagerot for the help during the software implementation.

Last but not least we would like to thank Magnus Wide and the rest of the Aros administration for allowing us to do our master thesis project at their company, and for giving us support during the development. We would also like to thank everyone else that has helped us at Aros and Chalmers with different tasks during the project.

Jakob Fjellström & Adam Maaninka, Göteborg June 11, 2014

Glossary

Below is a short description of the most frequently used abbreviations.

Name	Description
XMC	Main controller board
DMD	Motor controller board
SGB	Strain Gauge Board - Processes signal from load cell
OPC	Operator Presence Control/Dead Man's Switch
SPI	Serial Peripheral Interface - Communication protocol
LED	Light-Emitting Diode
CAN	Controller Area Network - Communication protocol
PDO	Process Data Exchange - Message type in CAN
NMT	Network Management - Message type in CAN
I2C	Inter-Integrated Circuit - Communication protocol
CAD	Computer-Aided Design
CM	Center of Mass
LHS	Left Hand Side of an equation
RHS	Right Hand Side of an equation
DC	Direct Current
AC	Alternating Current
PID	Proportional, Integral and Derivative controller
LCD	Liquid Crystal Display
LiPo	Lithium Polymer battery type
LP	Low-Pass filter
RC	Circuit with an resistance and capacitance (Low-Pass filter)
PCB	Printed Circuit Board

Contents

1 Introduction

THIS IS A MASTER THESIS project within the fields of Signals and Systems. The project specifically describes the challenges of implementing a mechatronic solution and control design on a shopping cart for the elderly.

1.1 Background

Elderly and disabled people might have a hard time when shopping due to the heavy lifting of shopping bags. A simple solution for this is to use a so called shopping trolley (Dramatenvagn) which is a type of shopping cart on two wheels pulled behind the user. The purpose of the shopping trolley is to facilitate the transportation by dragging the food rather than carrying it. This will of course remove the heavy lifting problem, however you still have to pull the shopping trolley up and down steep hills and stairs. For an elderly user this might be a huge problem.

1.2 Purpose

The problem where the elderly have to pull heavy trolleys up steep hills and stairs will be solved through an engineering point of view. The solution will ease the experience of shopping for the elderly.

1.3 Objective

The objective of the project is to develop a product based on the shopping trolley which solves the above stated problems. The product should assist the user pulling the shopping trolley based on user inputs. This solution has to be able to handle rough terrain such as steep hills, stairs and curbstones. Also, when pulling the trolley on flat surfaces it should feel comfortable.

1.4 Scope

This thesis work will cover the development of dynamic assistance systems for a shopping trolley. The work will include the whole development chain starting with the requirement specification, system design, modelling and simulation of the system and finally construction and analysis of a prototype.

The development work will be restricted in the following areas: marketing examination, interviews and demand checks on the product itself.

1.5 Method

The development process will be divided into several parts.

- Defining the primary functions that will be implemented in the new and improved shopping trolley. Examples of the functions are; electric drive on flat surfaces and stairs, load measurement and regeneration of the energy source.
- Use theoretical calculations to specify drive cycles of the shopping trolley.
- Model and simulate the system in *Simulink* to evaluate the overall performance.
- Dimension the actuators, energy sources and sensors based on simulations and calculations.
- Visualize the product through some form of CAD program.
- Construct a prototype.
- Design, implement, verify and validate a control system.

Optimized motor control, autonomous drive and other neat functions will be implemented if time allows.

1.6 Previous Work

Within the field of stair climbing there are a lot of different solutions which have to be taken into account. The stair climbing feature is used in many different products, from simple hand trolleys to advanced autonomous robots. This project however lies in between these areas. The product will be smart and powered by motors but is still driven by a user. Some of the previous work that is an inspiration to this work can be viewed in Figure 1.1. Observe that the final solution chosen for stair climbing and the rest of the system, is shown in Section 2.3.

Most of the references to previous work and other material presented in this report have been retrieved through the internet. Some references have been taken from compendiums from Chalmers University of Technology.

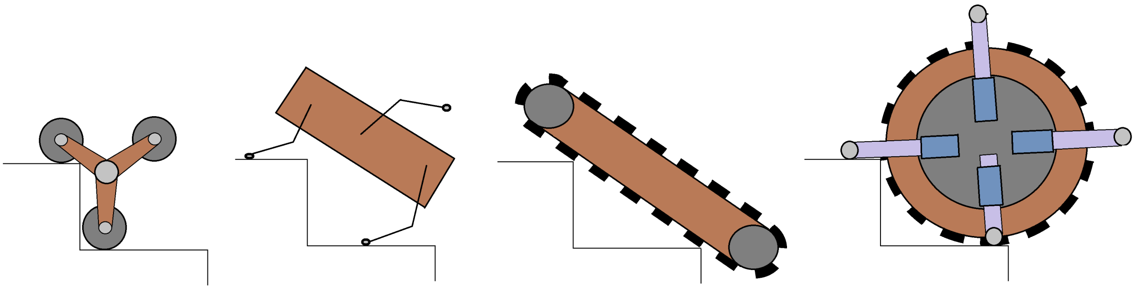


Figure 1.1: Different solutions for the stair climbing problem that has been examined in previous work. Starting from the left we have: Star wheel, Legs, Caterpillar tracks, Spokes/Big knobby tires.

1.6.1 Star wheel

One solution is often used on hand trolleys, and that is the tri-star wheel formation. It has three wheels that rotates about the center axis in a triangular shape (Fleming, 1969). When driving on flat surfaces the hand trolleys usually keeps two wheels on the ground. Another application for this solution is for small robots. According to (Moghadam and Ahmadi, 2007) the star wheel formation handles stairs quite well and fast in comparison to other solutions.

Another similar approach is to build a wheel with the same shape as the tri-star wheel but with the difference that everything is made in one piece. That is, the wheels on the edges of the triangle can not rotate individually, as studied by (Herbert et al., 2008). This solution also performs well in stair climbing. However, they point out that when moving forwards on flat ground an complex motion control algorithm is needed in order to get an adequate motion.

1.6.2 Legs

Another solution is using legs to climb the stairs. This has been tested by (Moore and Buehler, 2001), where they put three legs on each side of a rectangular robot. According to them, the robot had some problems when trying to climb. One problem was that it was likely to fall backwards. To avoid this the robot needed to move at low speed. Another problem was that sometimes the rear legs failed and slipped.

1.6.3 Caterpillar tracks

The caterpillar tracks are usually used on robots where good accessibility in difficult terrain is required. According to (Matthies et al., 1999) this solution performs well in an urban environment, and climbing stairs is no problem. However, one downside with the caterpillar tracks is that they have to be quite long and therefore use a lot of space.

1.6.4 Spokes/Big knobby tires

A lot of vehicles, for example motorcycles and electric wheelchairs, use the solution of big knobby tires to handle rough terrain such as stairs. Those vehicles often carry a lot of weight and need bigger motors. A similar solution is to use spokes to grip the stairs. (Jacovich, 2005) and (Wu, 2013) uses spokes attached to springs in order to keep the smoothness of wheels with the grip of spokes. The springs are compressed when facing ground.

1.7 Outline of the report

The report starts by covering the system design where the requirements and wishes for the project are presented. The next chapters capture the modelling, control design and simulation of the product to evaluate the system before the construction phase. The construction is divided into electrical and mechanical construction as well as software implementation. Finally the result is measured and analysed to evaluate if the final product fulfils the initial requirements.

2 System design

The system design describes the solution development process and the final chosen solution. The process starts with determining the demands and wishes for the trolley to withstand and finishes of with a sketch of the final solution.

2.1 Requirements and wishes

A requirement specification is made to be sure that the final product accomplishes the buyer's demands. The requirements and wishes are specified in Table 2.1 and 2.2 respectively. Both these tables are then used when comparing the different solutions. All solutions which do not meet the requirements are removed. To be able to use the wishes for comparation, the importance of the wishes are weighted. A solution which solves a high weighted wish is more attractive than a solution which solves a low weighted wish. The wishes with most importance is that the product is ergonomic and easy to use since the targeted users of the product are the elderly. It is also important that the solution is easy to implement due to the limited time of the thesis work.

Table 2.1: Requirements for the trolley

Requirement	**Description**	**Test Value**
R1	Assisting drive on flat surfaces	-
R2	Assisting drive in slopes	-
R3	Drive up stairs	-
R3.1	Straight staircase	-
R3.2	Spiral staircase	-
R4	Handle side walks	-
R5	The solution may not occupy more than a certain space	4 dm^3
R6	Handle a certain weight	40 kg
R7	Drive for a certain time	60 min
R8	Use of an electric motor	-
R9	Not complex usage	-
R10	Be able to turn easily	-

Table 2.2: Wishes for the trolley

Wish	**Description**	**Importance** (1-10 where 10 is highest)	**Weight**
W1	Be able to weigh the shopping content	3	0,048
W2	Easy recharging of battery	6,5	0,104
W3	Maximize torque at sensorless low speed	4	0,064
W4	Minimize torque ripple	4	0,064
W5	Display information to the user	7	0,112
W6	Autonomous drive	2	0,032
W7	Low total cost	5	0,080
W8	Ergonomic design	9	0,144
W9	Pretty design	7	0,112
W10	Easy to use	10	0,160
W11	Easy to implement	8	0,128
	SUM:	62,5	1,000

2.2 Solutions

When creating a total solution for a product it is wise to first make a function tree to get the partial functions of the product. In this case the main function is to give assistance when using the trolley. The main function is divided into partial functions which in turn can be further divided. The function tree with its partial functions is shown in Appendix A. For each function at the end of a branch, a list of possible solutions is made. The partial solutions are shown in Appendix B. The partial solutions are then combined into total solutions, see Appendix C. The total solutions which do not fulfil the requirements are removed. The remaining solutions are then compared and weighted to how well they fulfils the wishes. The top four total solutions in alphabetical order are:

- **Solution A** Uses one motor along with big wheels with rough texture to climb stairs. Angle measurements are used for control. A dead man's switch is used to ensure that help is wanted.

- **Solution C** Uses two motors along with big wheels with rough texture to climb stairs. Strain gauges and angle measurements are together used for control. A smart algorithm with existing sensors is used to ensure that help is wanted.

- **Solution J** Uses one motor along with wheels with spring spokes to climb stairs similar to (Wu, 2013). Strain gauges and angle measurements are together used for control. A smart algorithm with existing sensors is used to ensure that help is wanted.

- **Solution P** Uses two motors along with big wheels with rough texture to climb stairs. Spring spokes are added if the big wheels aren't enough. Strain gauge measurements are used for control. A dead man's switch is used to ensure that help is wanted.

2.3 Final solution

The final total solution chosen is solution **P**. Two motors are used to be able to turn easier. Big wheels with rough texture are used for climbing stairs along with additional spring spokes for better gripping of edges to ensure the stair climbing feature. Strain gauges are used to measure the force applied from the user to give adequate help. A dead man's switch is added to the handle to make sure that the motor doesn't drive when there is no user holding the handle. Figure 2.1 shows a CAD sketch of the final solution.

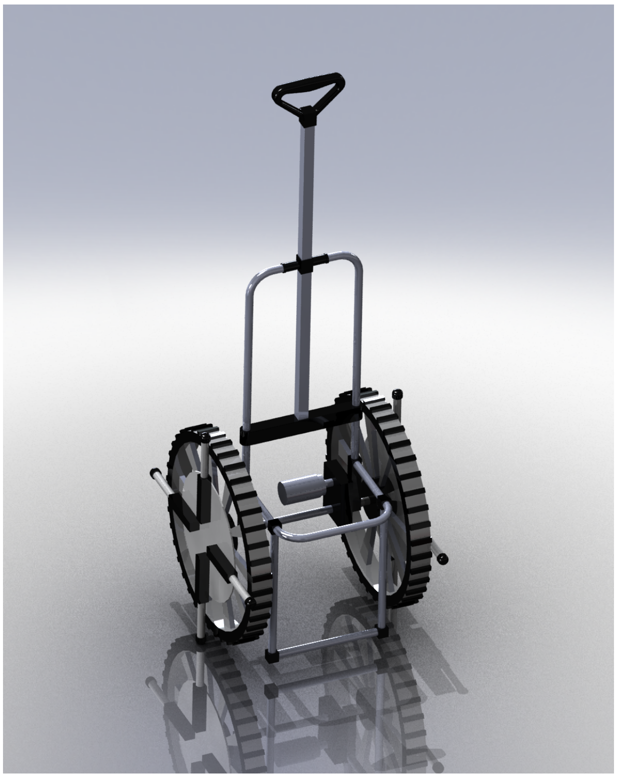


Figure 2.1: CAD sketch of the final solution.

3 Modelling

This chapter describes the modelling procedure of the project. First some simplifications are done to make the calculations easier. Then the Lagrange equation is used to retrieve a mathematical model of the shopping trolley (plant).

3.1 User Control System

An overall block diagram of the user holding the trolley and the trolley itself can be viewed in Figure 3.1.

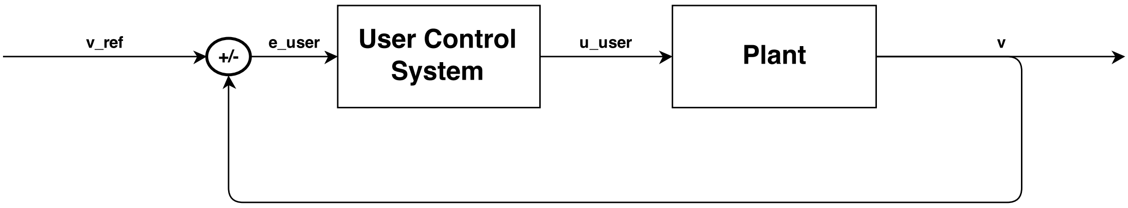


Figure 3.1: Block diagram of the user control system. $v_{ref,user}$ is the wanted velocity of the user pulling the trolley. v is the trolley velocity, e_{user} is the difference between the wanted velocity and the actual velocity, and u_{user} is the force applied by the user to the trolley.

Assuming that there are no other control inputs to the plant except for the force the user is pulling the trolley with u_{user}, the plant is simply describing a model of the trolley. The user control system block is describing the user's reactions. That is, when doing this type of simplification it is assumed that the force applied by the user is depending on the difference between the current velocity of the trolley and the velocity that is wanted by the user. Simply put, if the user is walking in a certain velocity and wants to increase the velocity, he or she will pull harder.

There is another way of looking at the user control system. If you look at the user's arm which is pulling the trolley, a thought is that it is possible to describe it using a spring and damper system. That is, at a close proximity to the trolley the arm is not pulling anything, however when a small step in any direction is made a force is generated. A larger step in the same direction would then imply an even larger force on the trolley. This can be compared to that the arm is stretching when the user is moving away from the trolley. However there is also a damper effect in place. Obviously the arm is not waving back and forth without stopping when walking with the trolley, and since there has to be some form of damping. A similar line of thought and the way to model the a human arm is also investigated in (Wang et al., 2010). This means that instead of the force (u_{user}) depending on the difference in velocity between the user and the trolley, it now depends on the displacement. A new block diagram can be viewed in Figure 3.2 describing this way of viewing the system.

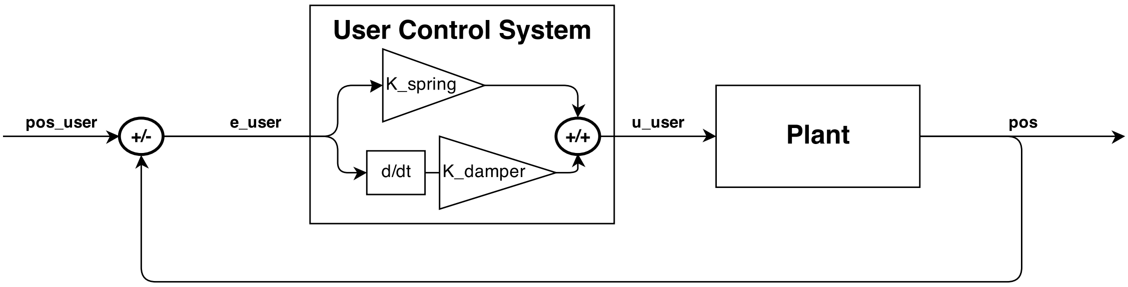


Figure 3.2: Block diagram of the user control system. pos_{user} is the position of the user pulling the trolley. pos is the trolley position, e_{user} is the difference between the position of the user and trolley, and u_{user} is the force applied by the user on the trolley.

Essentially this way of modelling the user and plant (through a spring and damper system) is going to be used later on in this report.

3.2 Trolley simplifications and sketch

To make the modelling of the plant (trolley) easier, see Figure 3.2, the following simplifications are made:

- The different parts of the trolley are simplified to blocks and point masses to easier handle the inertia.
- No slip is assumed between wheels and ground.
- No air resistance
- The z-position of the body frame in reference to the inertial frame is always zero. This means the trolley is at ground level constantly.

A sketch is made using the above information. Figure 3.3 and 3.4 shows the trolley with an above and side view respectively. The most important angles and positions are marked in the figures. X, Y and Z mark the inertial frame while x_b, y_b and z_b mark the body fixed frame. The pitch of the trolley is described with ψ which is the angle of the z_b axis relative to the Z axis. The angle of the wheels θ_r and θ_l are the rotations about the y_b axis. The yaw ϕ is the rotation of the body about the Z axis which is later described in terms of the wheel angles. All the initial constants and variables are described in Table 3.1.

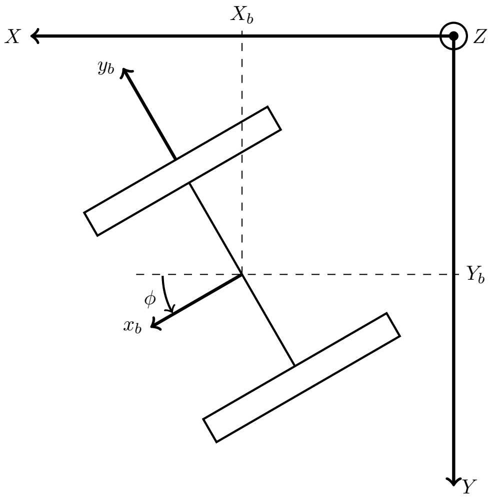


Figure 3.3: Sketch from above. X, Y and Z marks the inertial frame while x_b, y_b and z_b marks the body fixed frame. The yaw ϕ is the rotation of the body about the Z axis.

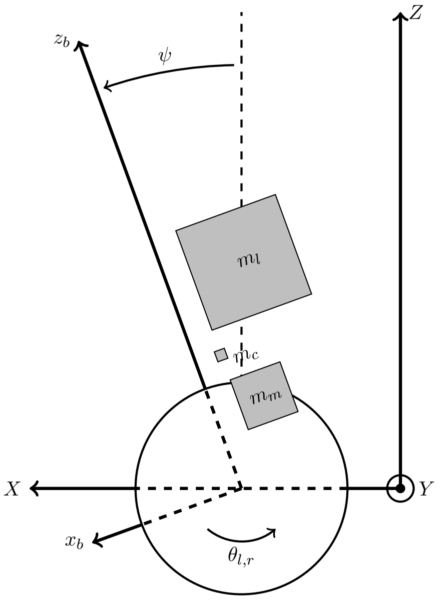


Figure 3.4: Sketch from side. The pitch of the trolley is described with ψ which is the angle of the z_b axis relative to the Z axis. The angle of the wheels θ_r and θ_l are the rotation about the y_b axis.

Table 3.1: Constants and variables

Name	Type	Description
$P_{bI} = [x_{bI}, y_{bI}, z_{bI}]$	[m]	Position of O of body frame in respect to inertial frame
Motor & Gear		
m_m	[kg]	Mass of motor and gear
D_m	[m]	Depth of motor
W_m	[m]	Width of motor
H_m	[m]	Height of motor
$\mathbf{P_{mB}} = [x_{mB}, y_{mB}, z_{mB}]$	[m]	Position of CM of motor in respect to body frame
i_r, i_l	[A]	Motor current (right, left)
V_r, V_l	[V]	Motor voltage (right, left)
I_m	[kgm^2]	Inertia of motor
K_t	[-]	Motor torque const.
K_b	[-]	Motor back EMF cosnt.
R_m	[Ω]	Motor resistance
Load		
m_l	[kg]	Mass of load
D_l	[m]	Depth of load
W_l	[m]	Width of load
H_l	[m]	Height of load
$\mathbf{P_{lB}} = [x_{lB}, y_{lB}, z_{lB}]$	[m]	Position of CM of load in respect to body frame
Chassis		
m_c	[kg]	Mass of chassi
$\mathbf{P_{cB}} = [x_{cB}, y_{cB}, z_{cB}]$	[m]	Position of CM of chassi in respect to body frame
Wheel		
m_w	[kg]	Mass of wheel
r_w	[m]	Radius of wheel
t_w	[m]	Thickness of wheel
$y_{rw,B}$	[m]	y-position of CM of right wheel in respect to body frame
$y_{lw,B}$	[m]	y-position of CM of left wheel in respect to body frame
Angles		
ψ	[rad]	Pitch of body frame in respect to inertial frame
ϕ	[rad]	Yaw of body frame in respect to inertial frame
θ_r	[rad]	Angle of right wheel
θ_l	[rad]	Angle of load wheel
Physical parameters		
$g = 9.82$	[m/s^2]	Gravity
f_w	[-]	Friction coeff. between wheel and ground
f_m	[-]	Friction coeff. between motor and chassi

3.3 Mathematical model of the Trolley

The mathematical modelling is divided into sections for easier reading. The sections describe the different parts of the modelling procedure. Variables and constants are described in Table 3.1. The differential equations achieved in this section results in the plant from Figure 3.1.

3.3.1 Inertial matrix

The first thing to calculate is the inertial matrix about the origin of the body fixed frame

$$I_B = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}. \tag{3.1}$$

The inertia about the CM (centre of mass) of each object of the body is calculated according to standard formulas of inertia as

$$\mathbf{I_m} = \begin{bmatrix} \frac{1}{12} m_m (W_m^2 + H_m^2) \\ \frac{1}{12} m_m (D_m^2 + H_m^2) \\ \frac{1}{12} m_m (W_m^2 + D_m^2) \end{bmatrix} \tag{3.2}$$

$$\mathbf{I_c} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.3}$$

$$\mathbf{I_l} = \begin{bmatrix} \frac{1}{12} m_l (W_l^2 + H_l^2) \\ \frac{1}{12} m_l (D_l^2 + H_l^2) \\ \frac{1}{12} m_l (W_l^2 + D_l^2) \end{bmatrix} \tag{3.4}$$

$$\mathbf{I_w} = \begin{bmatrix} \frac{1}{12} m_w (3r_w^2 + t_w^2) \\ \frac{1}{2} m_w (r_w^2) \\ \frac{1}{12} m_w (3r_w^2 + t_w^2) \end{bmatrix} \tag{3.5}$$

where $\mathbf{I_m}$, $\mathbf{I_c}$, $\mathbf{I_l}$ and $\mathbf{I_w}$ are the inertial vectors for the motor, chassi, load and wheels respectively. Bold characters denote a vector. The chassi is seen as a point mass and therefore has no inertia about its center of mass (CM). To create the elements of the inertial matrix Steiners theorems

$$I_{x'x'} = I_{xx} + m(d_y^2 + d_z^2) \tag{3.6}$$

$$I_{x'y'} = I_{xy} + m d_x d_y \tag{3.7}$$

are needed. The diagonal elements of the inertial matrix are calculated from (3.6) with the x, y, z values above as

$$\begin{aligned} I_{xx} = \; & [1,0,0]\cdot \mathbf{I_m} + m_m(y_{mB}^2 + z_{mB}^2) + [1,0,0]\cdot \mathbf{I_l} + m_l(y_{lB}^2 + z_{lB}^2) + \\ & [1,0,0]\cdot \mathbf{I_c} + m_c(y_{cB}^2 + z_{cB}^2) + 2 * ([1,0,0]\cdot \mathbf{I_w} + m_w y_{rwB}^2) \end{aligned} \tag{3.8}$$

$$\begin{aligned} I_{yy} = \; & [0,1,0]\cdot \mathbf{I_m} + m_m(x_{mB}^2 + z_{mB}^2) + [0,1,0]\cdot \mathbf{I_l} + m_l(x_{lB}^2 + z_{lB}^2) + \\ & [0,1,0]\cdot \mathbf{I_c} + m_c(x_{cB}^2 + z_{cB}^2) \end{aligned} \tag{3.9}$$

$$\begin{aligned} I_{zz} = \; & [0,0,1]\cdot \mathbf{I_m} + m_m(x_{mB}^2 + y_{mB}^2) + [0,0,1]\cdot \mathbf{I_l} + m_l(x_{lB}^2 + y_{lB}^2) + \\ & [0,0,1]\cdot \mathbf{I_c} + m_c(x_{cB}^2 + y_{cB}^2) + 2 * ([0,0,1]\cdot \mathbf{I_w} + m_w y_{rwB}^2). \end{aligned} \tag{3.10}$$

Assuming the masses are symmetrical about y_b there will only be one product of inertia which is

$$I_{xz} = m_c x_{cB} z_{cB} + m_l x_{lB} z_{lB} + m_m x_{mB} z_{mB} \tag{3.11}$$

according to (3.7). Since $I_{xy} = I_{yz} = 0$, all the components of the inertial matrix is found.

3.3.2 Rotational matrix and position definition

The rotational matrix is needed to be able to map the coordinates of the body fixed frame to the inertial frame (Boström, 2013). In this case it is assumed that the trolley has both wheels against the ground at all times, which means that there is no rotation about the x_b axis. The standard rotational matrices about the y_b and Z axis are multiplied to get the total rotational matrix from body to inertial frame

$$R_{BI} = \begin{bmatrix} \cos\phi\cos\psi & -\sin\phi & \cos\phi\sin\psi \\ \sin\phi\cos\psi & \cos\phi & \sin\phi\sin\psi \\ -\sin\psi & 0 & \cos\psi \end{bmatrix}. \tag{3.12}$$

where ψ and ϕ are the angles rotated about the y_b and Z axis respectively. The rotational matrix is used to define the position of the different masses in the inertial frame. The position of the origin of the body frame in the inertial frame is $\mathbf{P_{bI}} = [x_{bI}, y_{bI}, 0]$ which makes the position of for example the motor

$$\mathbf{P_{mI}} = R_{BI} \cdot \mathbf{P_{mB}} + \mathbf{P_{bI}} \tag{3.13}$$

with corresponding inertial positions for the other components. The denoted I or B describes if the variable or constant is in the inertial or body frame respectively. In a similar way

$$I_I = R_{BI} I_B. \tag{3.14}$$

3.3.3 Lagrangian

The goal with the mathematical model is to obtain the differential equations of the trolley. One method for finding the differential equations is using the Lagrangian equations

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \qquad i = 1,...,M \tag{3.15}$$

where L is called the Lagrangian (Boström, 2013), q_i is the generalized coordinates and Q_i is the generalized forces. The Lagrangian is built from the potential and kinetic energy of the system. The potential energy is

$$U = [0, 0, 1]M_{tot}g\mathbf{P_{CMI}} \tag{3.16}$$

where

$$M_{tot} = m_c + m_m + m_l \tag{3.17}$$

$$\mathbf{P_{CMI}} = R_{BI}\frac{1}{M_{tot}}(m_c\mathbf{P_{cB}} + m_m\mathbf{P_{mB}} + m_l\mathbf{P_{lB}}) + \mathbf{P_{bI}} \tag{3.18}$$

and the vector [0,0,1] extracts the height (Z) of the position vector $\mathbf{P_{CMI}}$. The kinetic energy is divided into the translational energy,

$$T_t = \frac{1}{2}m_w\dot{\mathbf{P}}^T_{\mathbf{rwI}}\dot{\mathbf{P}}_{\mathbf{rwI}} + \frac{1}{2}m_w\dot{\mathbf{P}}^T_{\mathbf{lwI}}\dot{\mathbf{P}}_{\mathbf{lwI}} + \frac{1}{2}m_{tot}\dot{\mathbf{P}}^T_{\mathbf{CMI}}\dot{\mathbf{P}}_{\mathbf{CMI}} \tag{3.19}$$

and the rotational energy

$$\begin{aligned} T_r = \ & \tfrac{1}{2}\dot{\theta}_r^2[0,1,0]I_w + \tfrac{1}{2}\dot{\theta}_l^2[0,1,0]I_w + \tfrac{1}{2}[0, \dot{\psi}, 0]^T I_B[0, \dot{\psi}, 0]+ \\ & \tfrac{1}{2}[0, 0, \dot{\phi}]^T I_I[0, 0, \dot{\phi}] + \tfrac{1}{2}n^2 I_m(\dot{\theta}_r - \dot{\psi})^2 + \tfrac{1}{2}n^2 I_m(\dot{\theta}_l - \dot{\psi})^2 \end{aligned} \tag{3.20}$$

which together with the potential energy (3.16) creates the Lagrangian

$$L = T_t + T_r - U. \tag{3.21}$$

There are also some equations of motion, see (Yamamoto, 2009), which can be used in order to eliminate the yaw angle ϕ and replacing it with a function of the wheel angles θ_r and θ_l. The left hand side (LHS) of (3.15) is then calculated with the generalized coordinates (ϕ, θ_r and θ_l) to obtain the differential equations.

$$\begin{bmatrix} \frac{d}{dt}\left(\frac{dL}{d\dot{\psi}}\right) - \frac{dL}{d\psi} \\ \frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_r}\right) - \frac{dL}{d\theta_r} \\ \frac{d}{dt}\left(\frac{dL}{d\dot{\theta}_l}\right) - \frac{dL}{d\theta_l} \end{bmatrix} \tag{3.22}$$

The right hand side of the Lagrange equations is given by the generalized forces (external forces) in the directions of the generalized coordinates. In this case there are at least four different forces. However, as a starting point only three of these forces are modelled.

Since there is three generalized coordinates which all are angles, their corresponding general forces are a sum of torques. This means the external forces need to be expressed in torques. The torque from the DC motors is described

$$T_{mr,l} = nK_t i_{r,l} \tag{3.23}$$

where n is the motor gear ratio, K_t is the motor torque constant and $i_{r,l}$ is the right and left motor current, see (Yamamoto, 2009). Using the standard DC motor equations and assuming that the DC motor inductance and friction inside the motor is negligible, the motor current

$$i_{r,l} = \frac{v_{r,l} + K_b(\dot{\psi} - \dot{\theta}_{r,l})}{R_m} \tag{3.24}$$

is calculated, where $v_{r,l}$ is the motor voltage, K_b is the back EMF constant of the motor and R_m is the motor resistance. The friction torque between the body and the motor is

$$T_{fmr,l} = f_m(\dot{\psi} - \dot{\theta}_{r,l}) \tag{3.25}$$

where f_m is the friction coefficient between body and motor. The friction torque between the wheel and ground is

$$T_{fwr,l} = f_w \dot{\theta}_{r,l} \tag{3.26}$$

where f_w is the friction coefficient between wheel and ground. From the above calculated torques, the three generalized forces

$$\begin{bmatrix} F_{\psi} \\ F_{\theta_r} \\ F_{\theta_l} \end{bmatrix} = \begin{bmatrix} -T_{mr} - T_{ml} - T_{fmr} - T_{fml} \\ T_{mr} + T_{fmr} - T_{fwr} \\ T_{ml} + T_{fml} - T_{fwl} \end{bmatrix} \tag{3.27}$$

(torques in this case) is found.

In order to get the differential equations on a form which is runnable in a simulation software (such as Simulink) it is favourable to have the equation on a standard form like $\ddot{x} = f(\dot{x}, x)$. Combining the right hand side (generalized forces) and left hand side (derivations of the Lagrangian) the differential equations

$$\begin{bmatrix} f_{\psi}(\ddot{\psi},\dot{\psi},\psi,\ddot{\theta}_r,\dot{\theta}_r,\theta_r,\ddot{\theta}_l,\dot{\theta}_l,\theta_l) \\ f_{\theta_r}(\ddot{\psi},\dot{\psi},\psi,\ddot{\theta}_r,\dot{\theta}_r,\theta_r,\ddot{\theta}_l,\dot{\theta}_l,\theta_l) \\ f_{\theta_l}(\ddot{\psi},\dot{\psi},\psi,\ddot{\theta}_r,\dot{\theta}_r,\theta_r,\ddot{\theta}_l,\dot{\theta}_l,\theta_l) \end{bmatrix} = \begin{bmatrix} F_{\psi} \\ F_{\theta_r} \\ F_{\theta_l} \end{bmatrix} \tag{3.28}$$

is achieved, which is not on the above stated standard form. That is, each equation contains the second derivative of each generalized coordinate. In order to get the equations in the standard form a factorization is required. Factorizing (3.28)

$$A(\dot{\psi},\psi,\dot{\theta}_r,\theta_r,\dot{\theta}_l,\theta_l) \begin{bmatrix} \ddot{\psi} \\ \ddot{\theta}_r \\ \ddot{\theta}_l \end{bmatrix} = \begin{bmatrix} F_{\psi} \\ F_{\theta_r} \\ F_{\theta_l} \end{bmatrix} \tag{3.29}$$

gives a new 3-by-3 matrix A, which is a function of lower order derivatives of the generalized coordinates. Multiplying both sided with the inverse of the A matrix

$$\begin{bmatrix} \ddot{\psi} \\ \ddot{\theta}_r \\ \ddot{\theta}_l \end{bmatrix} = A^{-1}(\dot{\psi},\psi,\dot{\theta}_r,\theta_r,\dot{\theta}_l,\theta_l) \begin{bmatrix} F_\psi \\ F_{\theta_r} \\ F_{\theta_l} \end{bmatrix} \tag{3.30}$$

gives the final differential equations which are usable in a simulation environment.

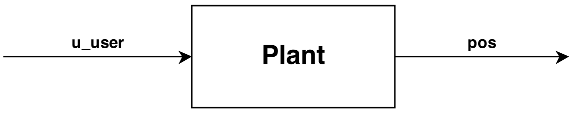


Figure 3.5: Block diagram of the user plant with input and outputs. *pos* is the output of the plant (position) and u_{user} is the input (force applied by the user on the trolley).

The plant block in Figure 3.5 represents the nonlinear model in (3.30), where

$$\begin{bmatrix} F_\psi \\ F_{\theta_r} \\ F_{\theta_l} \end{bmatrix} = f(u_{user}) \tag{3.31}$$

and

$$pos = f(\psi,\theta_r,\theta_l) \tag{3.32}$$

That is, the user force applied (u_{user}) is transformed into the three generalized torques through some simple geometry calculations. This transformation is needed because the user is applying the force along the handlebar and not directly on the wheels like the motor does. Also the position of the trolley in the inertial frame can be found through some simple calculations (summations) of the wheel angles over time.

4 Control Design

In the system design chapter (Chapter 2) the way to control the trolley was established. The controller is supposed to use either the angle measurements from a gyroscope and accelerometer, the pulling force measurements from strange gauges or a combination of both. However, for simplifications and due to time budget, only controllers based on force measurements is investigated.

4.1 Velocity control

A controller based on velocity is made using the force measurements from the strain gauges u_{user} as the reference from the user. However it does not make sense to compare the outputted velocity v from the plant directly to the input force from the user. So in order to compare the reference force from the user with the output velocity from the plant, some form of mapping (controlling) is required, see Figure 4.1.

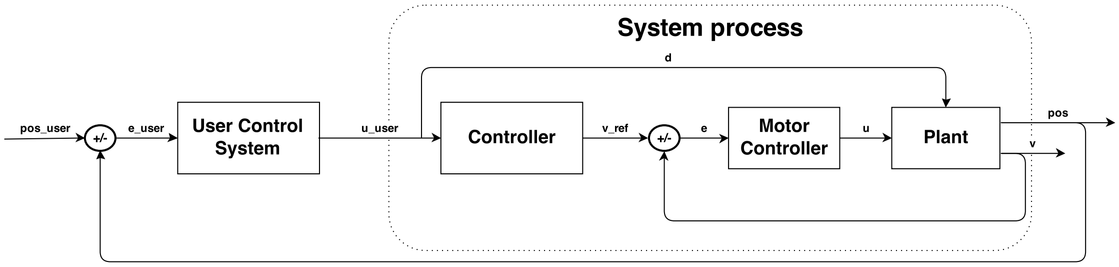


Figure 4.1: Block diagram of the entire system. pos_{user} is the position of the user pulling the trolley. v is the trolley velocity, e_{user} is the difference between the position of the user and that of the trolley, and u_{user} is the force applied by the user on the trolley. v_{ref} is the estimated wanted velocity of the user which is the mapped signal of u_{user} through the controller. The motor controller block is just representing a linear controller.

As noted in Figure 4.1 the Controller block is actually not closed loop within the system process, it is open loop. The loop is closed though through the user (human) interaction. Also in the controller block it is possible to shape the dynamics of the system. For instance a first order filter might be implemented here to give the user a smoother experience when pulling the trolley in addition to any proportional, integral or derivative effects wanted. The signal ”d”, which is actually the same as u_{user}, is affecting the plant directly in addition to being the input to the controller. That is, the d-signal is the force applied by the user on the trolley. The d-signal can be viewed as a disturbance or feed forward signal and is thus given the name ”d”.

The Motor Controller block which is added in the figure is only used to track the reference velocity v_{ref}. This block corresponds to the built in controller inside the ”real” motor, and it is of a PID type.

4.1.1 Proportional velocity controller

The first controller that is implemented has only a proportional gain. That is, the reference velocity that is given to the motor controller is proportional to the force applied by the user. This type of control makes it necessary to always have a user force greater than zero, since if the force is zero then the reference velocity also become zero in which case the trolley stops. In the case when a user starts pulling the trolley, starting from zero velocity, a large force is exerted on the trolley in the initial acceleration. This is then transformed into a reference velocity higher than the current velocity of the trolley and so the motor starts producing extra torque to increase the velocity. However when the desired velocity is reached the force the user was using to accelerate the trolley with is reduced since no more acceleration is wanted. This makes the reference velocity of the trolley lower as well, and the trolley reduces its velocity to an equilibrium point where the user is constantly pulling with a certain force. In other words there is some steady state error. And by that logic the steady state error should increase with increasing reference velocity from the user.

4.1.2 Integral velocity controller

Another type of velocity controller examined is an integral type controller. Lets look at Figure 4.1 again. Since v_{ref} is a velocity and u_{user} is a force, it is natural to think that the user force (u_{user}) is proportional to the wanted acceleration of the trolley. Then the reference v_{ref} increases or decreases proportionally against u_{user}. Simply put v_{ref} is the weighted sum of u_{user}, which means integral action. The belief is that this type of controller keeps on integrating the control signal until the user won't have to pull any of the load and therefore removes any steady state error. This feat is the thing that differentiate the integral controller from the other examined controllers.

4.2 Torque controller

Torque control is just what it sounds like. That is, controlling the torque of the motor based on force measurements. Simply put, the more force the user applies to the trolley the more help he/she will get. However, since this control method requires the user to always apply some force to get help, it has a built in a steady state error. The steady state error for the torque controller increases when a higher torque is wanted, since it is proportional to the measured force which in some sense is the reference/error that the controller wants to follow/correct.

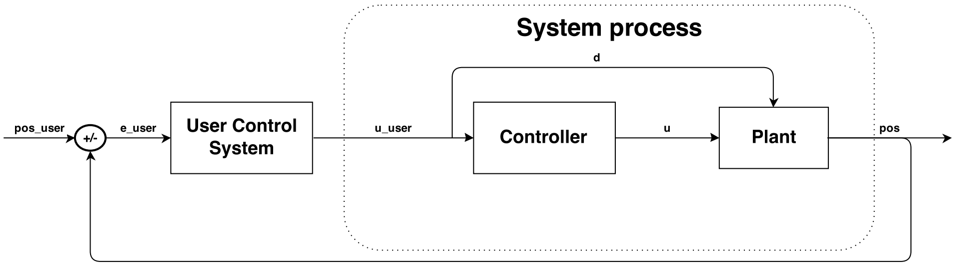


Figure 4.2: Block diagram of the entire system. pos_{user} is the position of the user pulling the trolley. e_{user} is the difference between the position of the user and that of the trolley, and u_{user} is the force applied by the user on the trolley. u is the control signal (torque in this case).

As can be noted in Figure 4.2, it looks a little different from the block diagram for the two velocity controllers in Figure 4.1. The difference is now that instead of following the velocity reference from the controller block, the motor controller block (not visible in Figure 4.2) now just applies the torque that the controller block outputs directly to the plant.

5 Simulation

Based on the model described in Chapter 3 simulations are made in Matlab/Simulink, see Appendix D. The model of the trolley and motor is somewhat correct, however since the models and simulations are made before the actual prototype is built, they do not fully represent the final product. However the major characteristics are there and the simulations is foremost used for evaluating the control strategies of the different controllers described in Chapter 4 before implementing them in the real plant.

5.1 No controller

As a start only simulation using the model of the plant together with the model of the user is made. A step response is simulated where the user velocity starts at 0 m/s and ends at 1.4 m/s. Added to the velocity of the user is a sine wave with frequency 1.8 Hz and amplitude 0.1 m/s in order to simulate the steps taken by the user pulling the trolley. The velocity signal is then run through a first order filter with a time constant of 0.1 s and then integrated to get the position of the user. The difference in position between the user and the trolley is then the input to the user control system which is modelled by a spring/damper system, as described in Section 3.1. Out from the user control system block is the force which is applied to the plant and also measured. In this case the motor is turned off and only the user force is acting on the plant.

As can be seen in Figure 5.1 the velocity of the user is oscillating as intended which affects the pulling force. The velocity of the trolley is also oscillating however at a lower amplitude due to the damping effect in the user control system. That is, the damping comes from the arm of the user pulling the trolley. The amount of force required to pull the trolley at 1.4 m/s is about 30 N, which seems reasonable.

5.2 Proportional velocity controller

The proportional velocity controller described in Section 4.1.1 is evaluated. The result can be viewed in Figure 5.2.

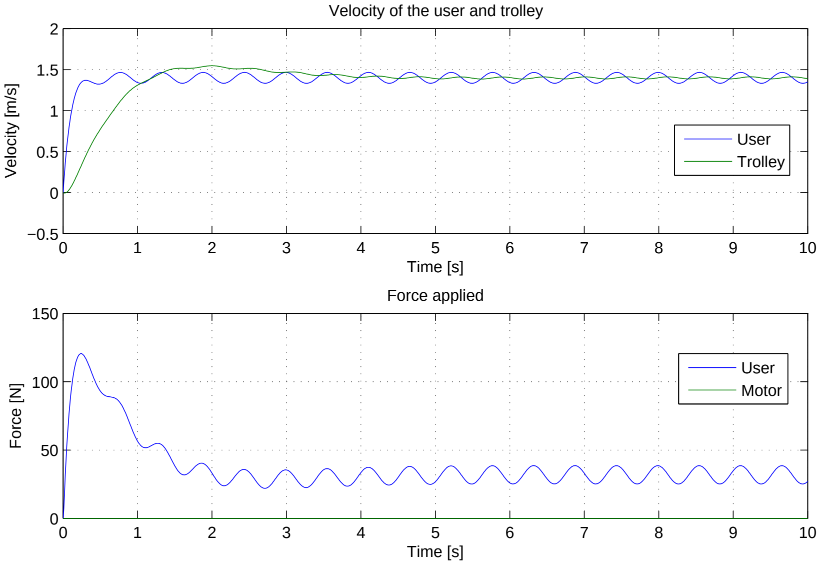


Figure 5.1: Simulation results when the controller and motor is turned off. The only thing pulling the trolley is now the force directly applied by the user.

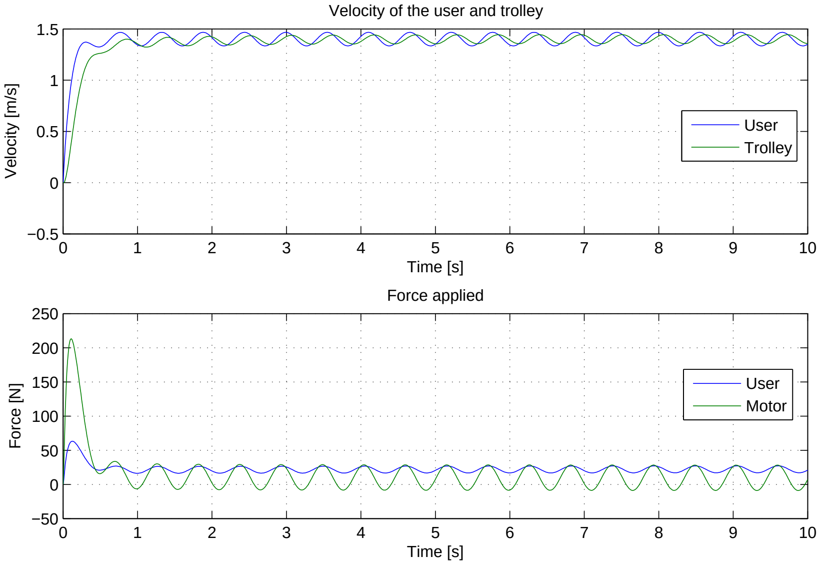


Figure 5.2: Simulation results when the controller is set on velocity control using a PD-controller. Both the user force and the torque generated by the motor affects the system.

When comparing the resulting graph in Figure 5.2 with the results when no controller is active, Figure 5.1, it is deducted that the force applied by the user is now lower. It is measured to be about 20 N from earlier 30 N in steady state, which is about a 30 % reduction. The user is also getting help in the initial acceleration part where the force required is almost halved. However, it is also visible in the figure that the control signal (motor force) is oscillating a lot. This type of behaviour might not be desired.

Another simulation with the same controller parameters is made, this time a first order filter is implemented in the controller in order to make the control signal oscillate less. The result can be viewed in Figure 5.3.

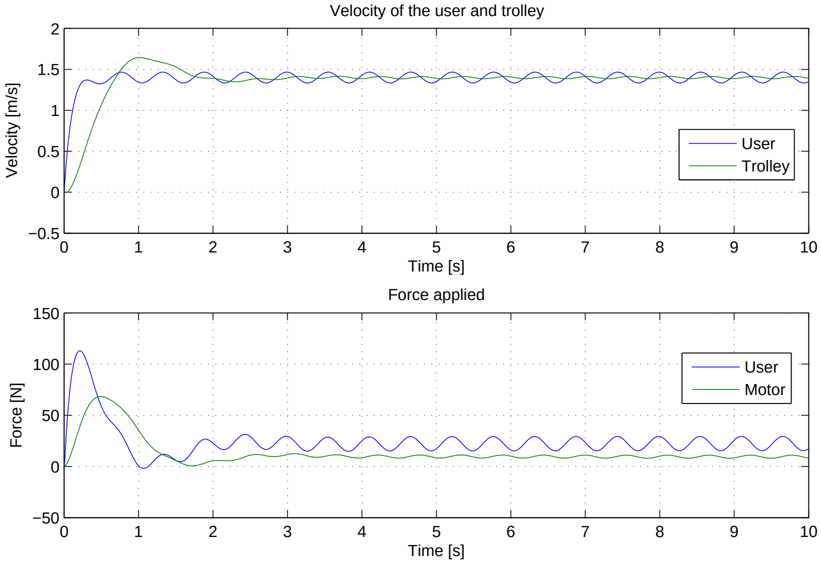


Figure 5.3: Simulation results when the controller is set on velocity control using a PD-controller. Both the user force and the torque generated by the motor affects the system. A first order filter is added in the controller block to get a more stable control signal.

As can be noted in Figure 5.3 the control signal is oscillating less, however at the expense of not giving as much help in the early acceleration part. As expected the user control signal never reaches zero, which means the user constantly needs to add force in order to get any help from the motor.

5.3 Integral velocity controller

The second controller that is evaluated is also a velocity controller. This controller however has no proportional or derivative part, but only an integral part. The results can be viewed in Figure 5.4.

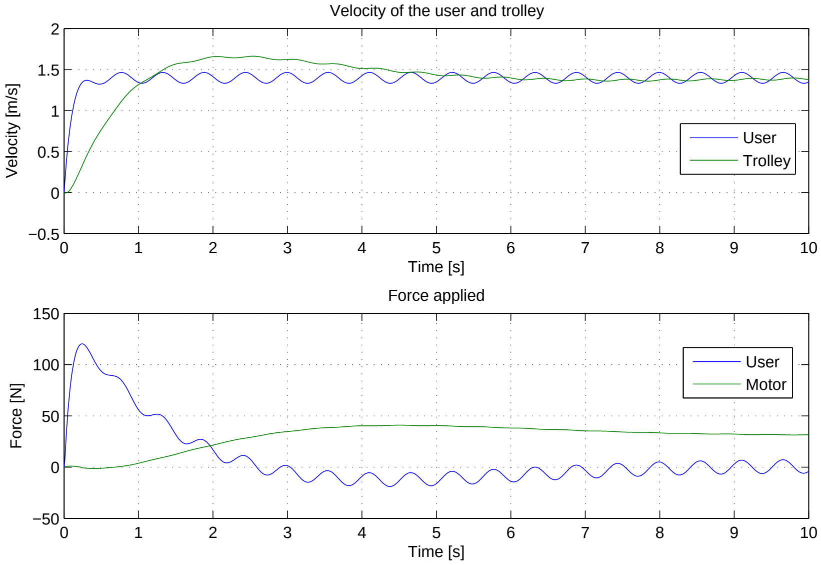


Figure 5.4: Simulation results when the controller was set on velocity control using a I-controller. Both the user force and the torque generated by the motor affects the system.

Here the controller is smooth from the beginning and no first order filter is needed as for the earlier controller discussed in Section 5.2. It is noticed however that the user is not getting a lot of help in the initial acceleration phase, but as time progresses the user force is tending towards 0 N. This means that after a while the user doesn't need to pull at all, the trolley will hold its velocity anyway.

When doing these simulations, it is also noted that if measurement noise is added to the force measurement (in the form of an offset), the integral action of the velocity controller makes the trolley increase its velocity and closes in on the user pulling it. This can be viewed in Figure 5.5.

This behaviour results in a negative force from the user (that is, the user pushes the trolley back continuously). This is something that is also noticed when testing the final prototype. When the force measurement sensor is not calibrated properly, which means it has an offset, the integral action keeps on increasing the velocity of the trolley as later discussed in Section 9.2.1 about the integral velocity controller.

5.4 Torque controller

The last type of controller that is evaluated is the torque controller. The results can be viewed in Figure 5.6

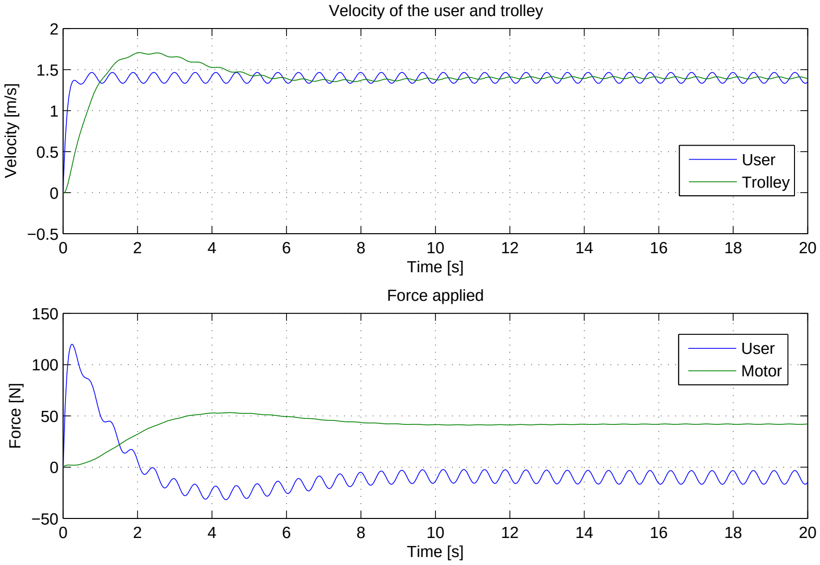


Figure 5.5: Simulation results when the controller is set on velocity control using a I-controller. Both the user force and the torque generated by the motor affects the system. Here an offset was added to the measurement of the user pulling force.

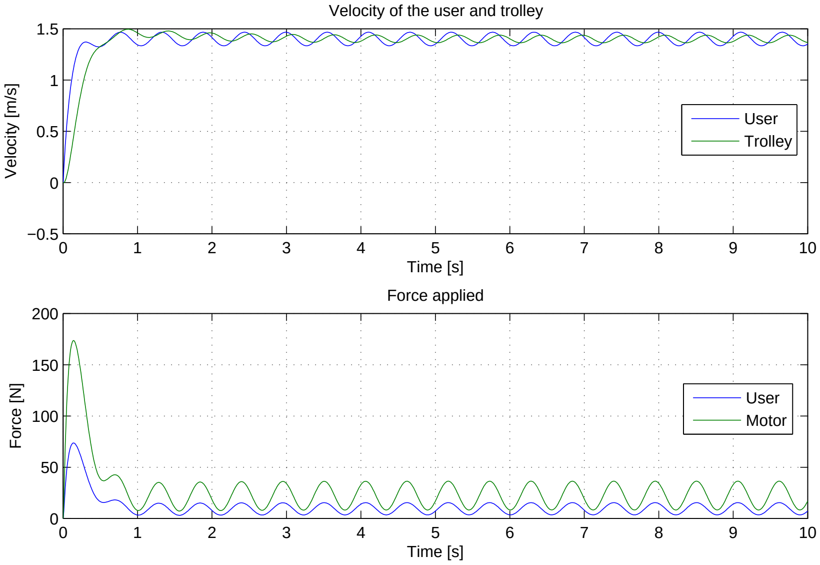


Figure 5.6: Simulation results when the controller is set on Torque mode using a P-controller. Both the user force and the torque generated by the motor affects the system.

The resulting graph shows that the motor torque is proportional to the user force. This makes sense and is expected. However this control method, like the proportional velocity controller, requires that the user always applies a force to the trolley unlike the integral velocity controller as discussed in the previous section.

6 Electronic construction

During the project a lot of electronic construction is made. In this chapter the different parts and their connection is explained. The electronic system is described followed by the force measurement implementation. At the end the user interface is described.

6.1 System description

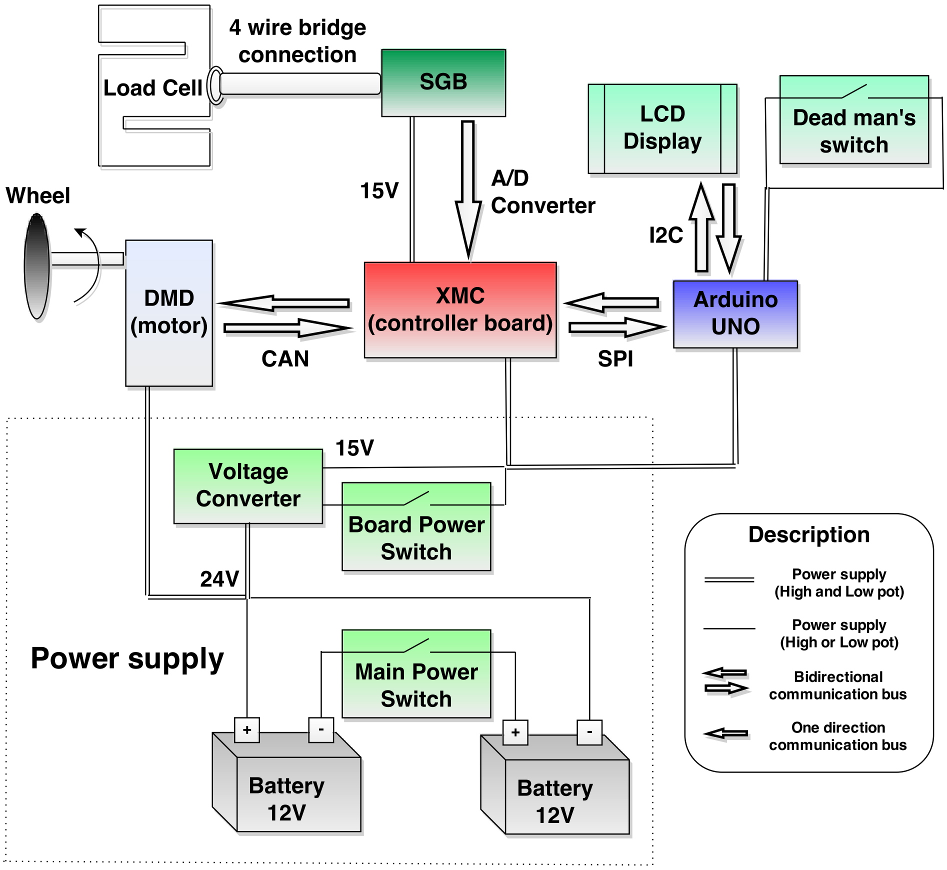


Figure 6.1: A description of the electrical system.

Figure 6.1 displays an overall system description. The red board in the middle is the XMC board which is the centre piece of the system. It is in this board the controllers are implemented, that is the controller explained in Chapter 4. The XMC board communicates with the rest of the system using different communication protocols. The sensor (load cell that measures the force from the user) has the simplest form of communication, that is the XMC just reads an analogue value from the SGB (Strain Gauge Board) and supplies it and the load cell with power.

The Arduino pretty much just passes through the information about the controller parameters and states of the system from the XMC to the display. It also contains a menu in which it is possible for the user to select which information to be viewed on the display, as well as letting the user change the controller parameters through buttons located on the display. The communication is here done by using SPI between the XMC and Arduino, and I2C between the Arduino and the LCD display.

The XMC passes the control signal through a CAN bus to the DMD which is the motor controller board. The DMD has a simple PID controller implemented for controlling the motor based on the reference given by the XMC.

Everything is powered through the power supply constellation which consists of two 12 V, 50 C and 6000 mAh LiPo batteries connected in series with a main power switch in between. The motor is directly connected to the 24 V power supply while the other boards are supplied with a 15 V signal. This is due to the maximum allowed voltage on the boards. This means that the 24 V signal needs to be converted down to 15 V. The 15 V signal also has a power switch, which is actually a switch connected to a relay, that turns the power to the boards on/off separated from the power to the motor.

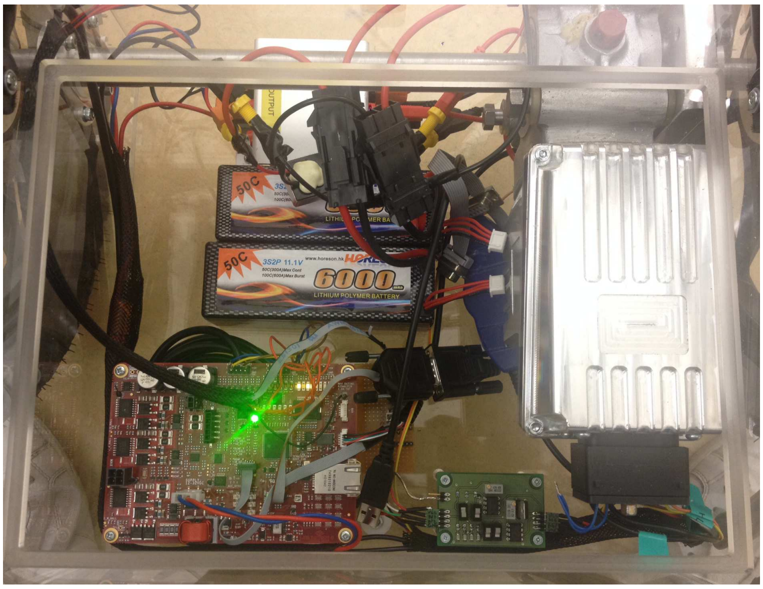


Figure 6.2: In the left bottom corner is the red XMC board, the bottom right is the green SGB board. Inside the metallic box to the right is the motor control board.

Figure 6.2 shows the electronic system constructed. In the left bottom corner of Figure 6.2 is a red board called XMC, which contains the controller implementations. The small green board in the lower right is the SGB board which filters and amplifies the signal from the load cell. Inside the motor chassis in the upper right corner is the motor control board (the DMD board). It contains the control algorithm for velocity tracking used by the motor. Above the red board lies two batteries and a metallic box hidden below the red and black cables. They form (together with the switches that aren't visible in the picture) the power supply constellation. The fourth board is an Arduino (not shown in the picture) which operates as user interface. The Arduino is further described in Section 6.3.2

6.2 Measuring force

In Chapter 2, system design, it is decided that the trolley is going be controlled by force measurements from strain gauges. Strain gauges are very sensitive resistors which are applied to the surface of the object to which the force will be applied. In this case they are applied to the rod below the handle. When a force is applied the rod will slightly deform and cause the resistor to deform as well. The deformed resistor will have a higher resistance than before. The difference in resistance is measured and later converted to a force in the software.

A more reliable solution is to use a load cell which is a metal component with strain gauges already applied to it. That way the insecurities of attaching the strain gauges are removed. Also one can be certain that the metal itself has the abilities suitable for the specific purpose. One drawback of the load cell is that the trolley has to be rebuilt. The rod has to be split in two and the load cell attached in between the two parts.

6.2.1 Placement of strain gauge

The placement of the strain gauges are important. Different positions give different performances. For this project the strain gauges is placed according to Figure 6.3. The objective is to measure the force component in the direction of the rod. When the force F is applied, strain gauges R_2 and R_4 will deform equally and R_1 and R_3 will deform virtually nothing.

The strain gauges are connected to a Wheatstone bridge according to Figure 6.4 to be able to measure the difference in strain. The measured difference is

$$V_{out} = \frac{R_1 R_3 - R_4 R_2}{(R_3 + R_2)(R_1 + R_4)} V_{in}. \tag{6.1}$$

which can be used to make sure the strain gauges are placed correctly. To accomplish the previously mentioned objective, V_{out} needs to measure a force in the direction of the rod but be resistant to pure torques. When a force in the direction of the rod is applied $R_2 = R_4 = R + \Delta_R$ and $R_1 = R_3 = R$. Since the deformation of the aluminium rod is small the deformation of the resistors resolve in a resistance difference Δ_R of only ±100

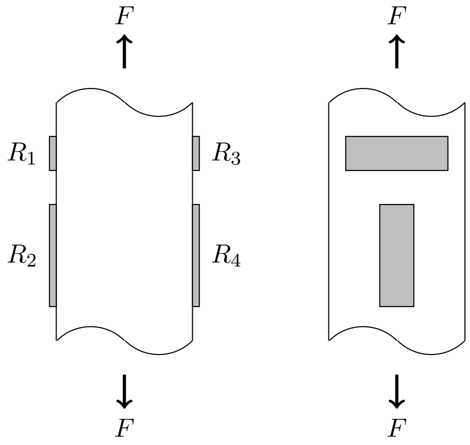


Figure 6.3: Placement of strain gauges

mΩ. The strain gauge nominal resistance R is 350 Ω and the input voltage V_{in} is 5 V. Used in (6.1)

$$V_{out} = \frac{R^2 - (R + \Delta_R)^2}{(R^2 + \Delta_R)(R^2 + \Delta_R)} V_{in} = \frac{-2R\Delta_R - \Delta_R^2}{(R^2 + \Delta_R)^2} V_{in} \approx -0.71 \text{ mV} \tag{6.2}$$

which means $V_{out} \neq 0$. However, when the rod is bent $R_2 = R + \Delta_R$, $R_4 = R - \Delta_R$ and $R_1 = R_3 = R$ which gives

$$V_{out} = \frac{R^2 - (R - \Delta_R)(R + \Delta_R)}{(2R + \Delta_R)(2R - \Delta_R)} V_{in} = \frac{\Delta_R^2}{4R^2 - \Delta_R^2} V_{in} \approx 0.1 \ \mu\text{V}. \tag{6.3}$$

This result means that torques affecting the rod will be approximately zero in comparison with a force in the direction of the rod which is the wanted behaviour.

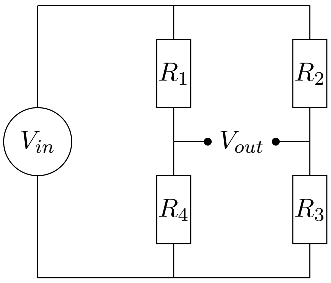


Figure 6.4: Strain gauge placement in Wheatstone bridge

If using a load cell the strain gauges inside the metal component is already connected in a Wheatstone bridge which means that the load cell will only measure force in a specified direction. The inputs to the load cell are V_{in} and ground and the outputs are the two nodes creating V_{out}.

6.2.2 Signal processing

The output signal from the Wheatstone bridge is in the range of ±1 mV. The analogue inputs of the XMC board is in the range of 0 − 6 V with an A/D converter with 4096 steps which results in a step size of 6000/4096 = 1.4 mV. It is obvious that the output signal V_{out} needs to be processed to get a suitable range.

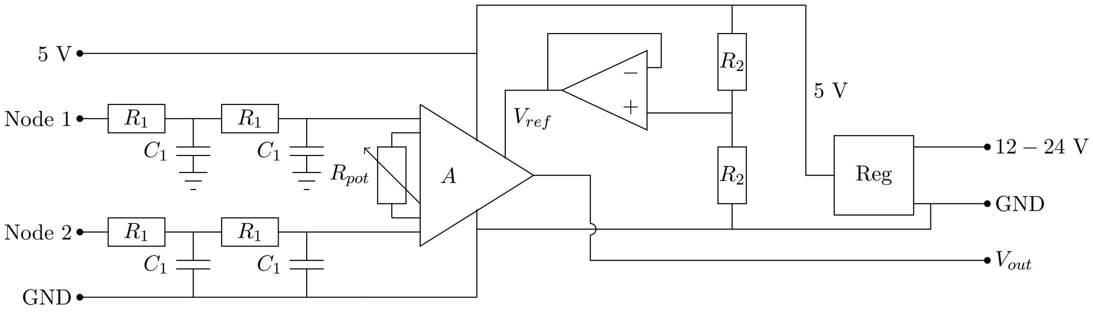


Figure 6.5: Circuit of signal processing board. The left side is connected to the Wheatstone bridge. The right side is connected to the XMC board.

Figure 6.5 shows the circuit of a developed signal processing board. The purpose of the board is to filter, amplify and offset the output signal from the Wheatstone bridge so that the signal offset is 2,5 V with a range of ±500 mV. Four switches are also added to the circuit (not shown in the figure). Two of them are used to turn the filters on and off, one is used for choosing the input voltage to the Wheatstone bridge (5 or 12 − 24 V) and the last one is used to turn balancing of the Wheatstone bridge on and off. The balancing of the Wheatstone bridge (not shown in the figure) is used to set the difference between Node 1 and 2 to zero when no force is applied. This is useful for example if the gluing of the strain gauges isn't perfect.

Low Pass Filtering

The low pass filters are built from two RC-circuits in series. The filter circuits are designed to suppress frequencies above 10 Hz. With $R_1 = 100$ kΩ and $C_1 = 100$ nF, the LP filter is simulated using LTSpice, the resulting bode plot is shown in Figure 6.6 and the transient response in Figure 6.7.

The bode plot confirms the values of the resistance and the capacitance. At 10 Hz the amplification is -5.9321 dB which means the cut off frequency is slightly lower. At higher frequencies the suppression is about 40 dB per decade which matches that of a second order Butterworth filter. From the transient response it is also noticed that the time constant is about 30 ms and the rise time about 58 ms. Altogether it is assumed that force changes from the user will be slower than 10 Hz. Higher frequencies is thought of as noise and will be suppressed.

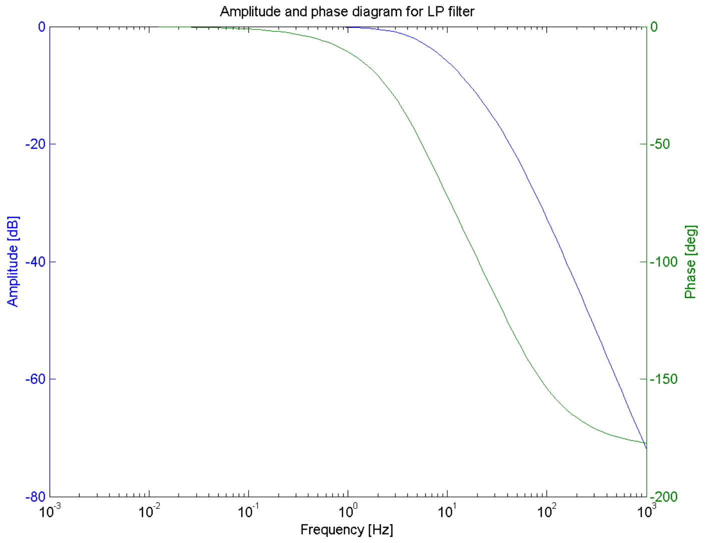


Figure 6.6: Bode plot of low pass filter

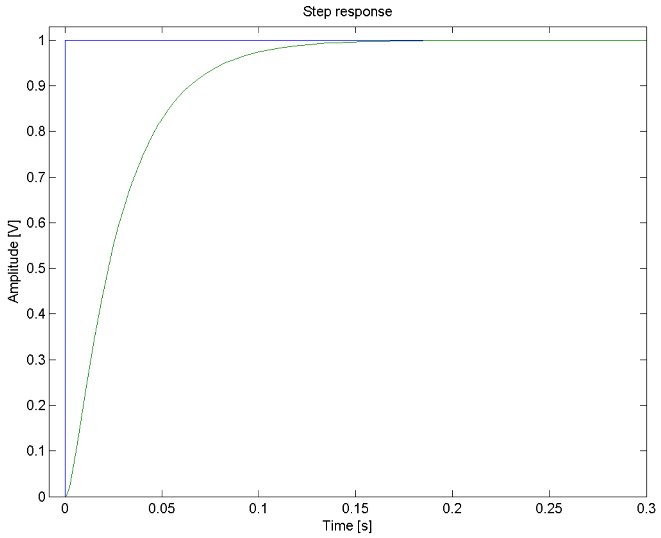


Figure 6.7: Step response of low pass filter

Amplification

As previously mentioned the output signal from the Wheatstone bridge is a couple of mV while the analogue input of the XMC board has the range of 0-6 V. In Figure 6.5 the difference between Node 1 and Node 2 (after being filtered) is amplified with a instrumental amplifier. The resistor connected to the amplifier specifies the amplification. By using a potentiometer of 200 Ω the amplification will be variable with a range of about $300 - 10000$ times according to the data sheet for the component INA128. The highest resistor value determines the lowest possible amplification.

Offset

The offset is needed so that the output value doesn't vary about zero. In the case that the the signal varies about zero only positive signals would be measurable, since the voltage range of the A/D converter is 0 to 6 V. To match the analogue input to the XMC board the wanted offset is 2,5 V. This can be managed by feeding the reference port of the instrumental amplifier V_{ref} with 2,5 V while the supply voltages V_+ and V_- is 5 V and ground respectively. The 6 V signal is regulated from the $12 - 24$ V input voltage to the circuit. From the 5 V signal, V_{ref} is created using a voltage divider in series with a voltage buffer.

6.2.3 Implementation and results

The force measurements is realized in a couple of ways. The first implementation is strain gauges used with the placement and Wheatstone bridge configuration as mentioned above. To measure the strain a prototype signal processing board (SGB) is built according to Figure 6.5 using stripboard. The SGB is shown in Figure 6.9 (the left picture). This configuration is used to measure the force needed to pull the trolley with a 20 kg load in different slopes. From these measurements the maximum needed motor torque can be estimated. The noise from the strain gauges with the prototype SGB is shown in Figure 6.8, top graph. The noise has an amplitude of ±10 N and has an offset depending on temperature and plastic deformation in the rod. This means the signal is too noisy to be used as reference to the controller since the signal to noise ratio is too low. If using this configuration the noise may be interpreted as a signal which could cause the trolley to move without a user input signal.

To get rid of the signal offset a load cell is tested. While connected to the prototype SGB the noise is measured (Figure 6.8, middle graph). The result is much more stable with an amplitude of ±0.5 N. There is no plastic deformation (or at least small enough to not be visible on the measurements) and since the strain gauges are integrated in a metal construction they are more resistant to temperature changes. At this point the noise is mostly from the prototype SGB.

Using Eagle an improved circuit and PCB layout is developed, shown in Figure E.1 and E.2 in Appendix E. From those, a printed circuit board is manufactured, Figure 6.9 (the right picture), to remove the unnecessary noise of the prototype signal processing board. The manufactured SGB gave the noise shown in Figure 6.8, lower graph. The resulting noise is at most ±0.05 N which is more than enough. A signal of only 0.5 N is easy to distinguish. Hence, this is used as the final configuration.

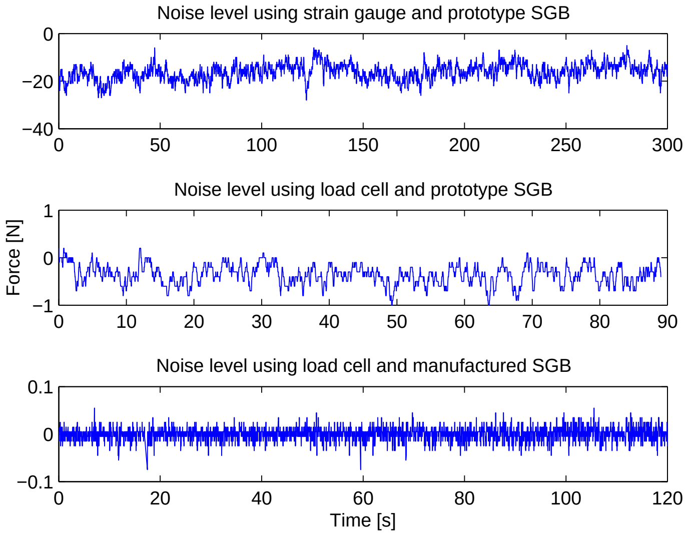


Figure 6.8: Noise measurement of the different force measurement configurations.

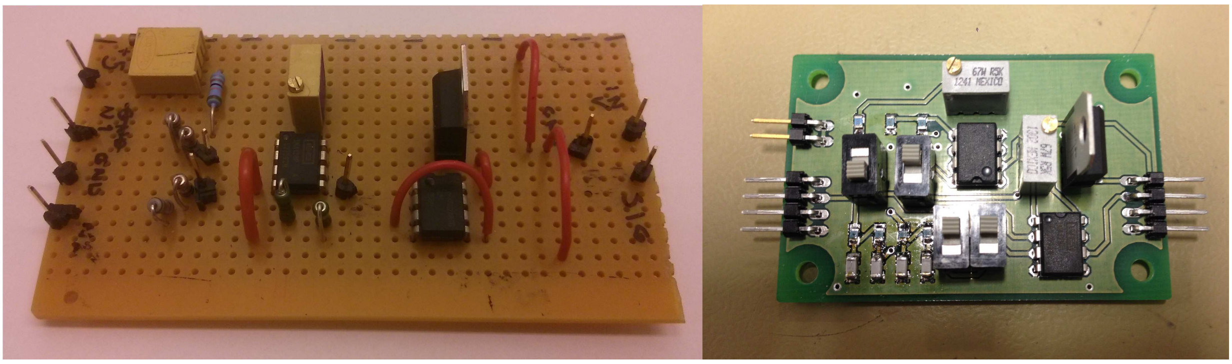


Figure 6.9: Prototypes of developed signal processing board. At the left is the first prototype board and to the right is the newer PCB.

6.3 User interface

Important parts of the trolley are the safety when driving the trolley and some way of getting information to the user. These functions are described in this section.

6.3.1 Dead man's switch

What happens if the user suddenly drops the handle, or tries to lift the trolley straight up without holding the handle? Should the motor keep on turning in these situations or should it stop? These are the main questions that needs to be answered in the safety aspect of the trolley. That is, how to ensure that the trolley won't run over the user if he/she looses control of it. The result of the solution elimination from Chapter 2 is to use a so called dead man's switch or another name for it Operator Presence Control (OPC). The OPC can be viewed in Figure 6.10.

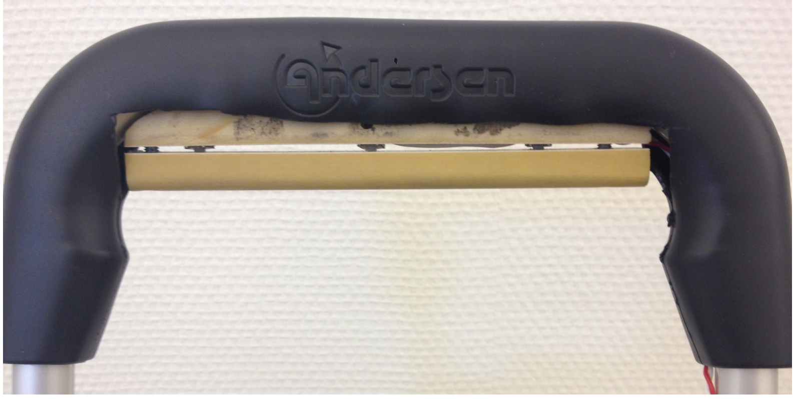


Figure 6.10: The dead man's switch which is needed to be activated in order for the motor to run.

The good thing about the OPC is that it is easy to implement in hardware and in software perspective. The OPC is constructed by two pieces of plastic put together with screws. In between the two pieces three switches is placed. When the user squeezes the OPC the switches activate and the two cables leading to the Arduino is short circuited, which allows easy detection. Simply put, if the user squeezes the handle the motor runs, if the handle is dropped or let go the motor stops.

6.3.2 Arduino

In order to display information to the user a display shield together with an Arduino is used which can be viewed in Figure 6.11.

The Arduino communicates with the XMC board via SPI. An advantage with using the Arduino is that it comes with plug and play ready solutions for most applications. However, when using SPI with an Arduino the plug and play solutions only supports the Arduino as a master. In this case the Arduino is set as a slave for which a custom SPI code is implemented. Another problem when using the SPI with the Arduino is that the Arduino uses 5 V while the XMC uses 3.3 V which requires a level shifter.

6.3.3 Power switches

As described in earlier sections there are two power switches and four LEDs as can be viewed in Figure 6.12. The right most switch (black with a pink light) is used for cutting off the power to every component in the system. The right most switch (metallic) is used for only rebooting the control system. The LEDs come in pairs of two. The right most pair displays the level of the battery, that is if it needs recharging. The right most LEDs

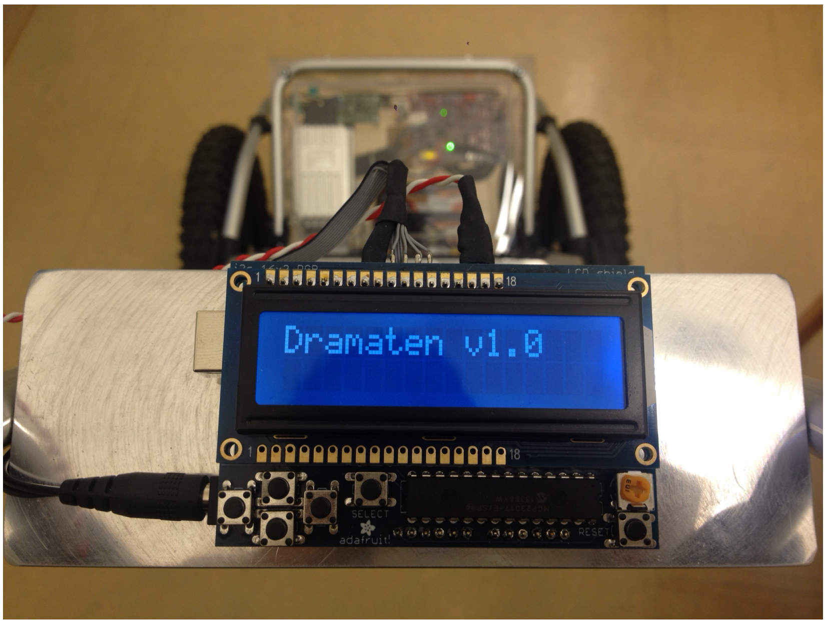


Figure 6.11: The Arduino together with a display shield. The user can set the control parameters online by using the Arduino which communicates through SPI to the XMC board.

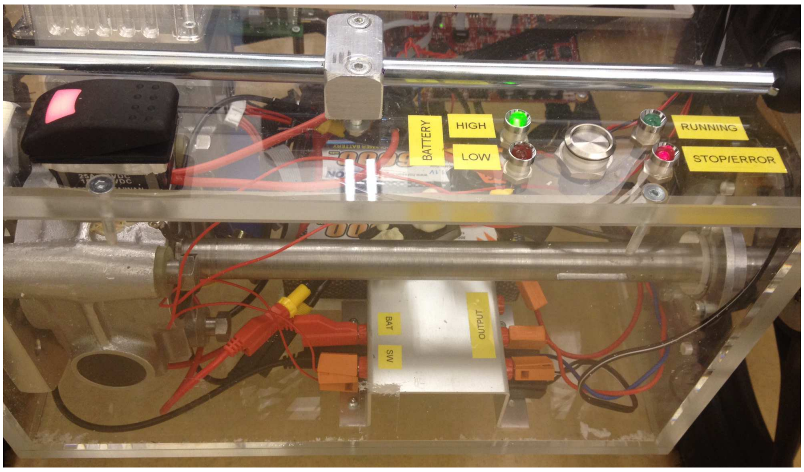


Figure 6.12: The led lights displays the current battery level and control mode. The switch on the right switches on/off the XMC and SGB boards. The switch to the left switches all power on/off.

tells the user if the trolley is in running mode and reacts to forces in the rod leading up to the handle or if it is stopped/error mode and doesn't supply any torque.

7 Mechanical construction

This chapter explains the different parts of the mechanical construction of the prototype as well as the procedure of finding the motor properties.

7.1 Motor design

The design of the motor and potential gearbox is important to the project to get an appropriate amount of help when for example going up a steep hill. In this section calculations and measurements are compared and evaluated to conclude what motor properties are needed.

7.1.1 Torque calculations

This section contains calculations to estimate the motor torque required to drive the trolley in different slopes. Since the mathematical model developed in Chapter 3 does not include inclination of the slope, these calculations uses a simpler model with some added equations. The total torque T is

$$T = T_w + T_b \tag{7.1}$$

where T_w is the sum of the torques needed to rotate the wheels and motor and T_b is the torque needed to drive the body up the slope. The two different torques are calculated as

$$T_w = \frac{(I_m + I_w)\ddot{x} + f_m\dot{x}}{r_w} \tag{7.2}$$

and

$$T_b = (f_w\dot{x} + m_{tot}\ddot{x} + m_{tot}g\sin\alpha)r_w, \tag{7.3}$$

where the variables and constants are described in Table 7.1.

The total torque is calculated according to a predefined velocity and acceleration behaviour to get a good estimate. The velocity curve is created to get a time constant of approximately 0.5 seconds. The velocity and acceleration is shown in Figure 7.1 to the left.

In the right graph of Figure 7.1 the total torque is shown. The torque is calculated at several different degrees of inclination with the velocity and acceleration behaviour as previously described. The torque behaviour follows that of the acceleration but reaches

Table 7.1: Constants and variables used while calculating torque required.

Name	Type	Description
I_m	[kgm^2]	Inertia of motor
I_w	[kg^2]	Inertia of wheels
f_m	[-]	Friction in motor
f_w	[-]	Friction between wheel and ground
r_w	[m]	Wheel radius
m_{tot}	[kg]	Total mass of trolley
α	[°]	Degrees of road inclination
$\dot{x}$	[m/s]	Velocity of trolley
$\ddot{x}$	[m/s^2]	Acceleration of trolley

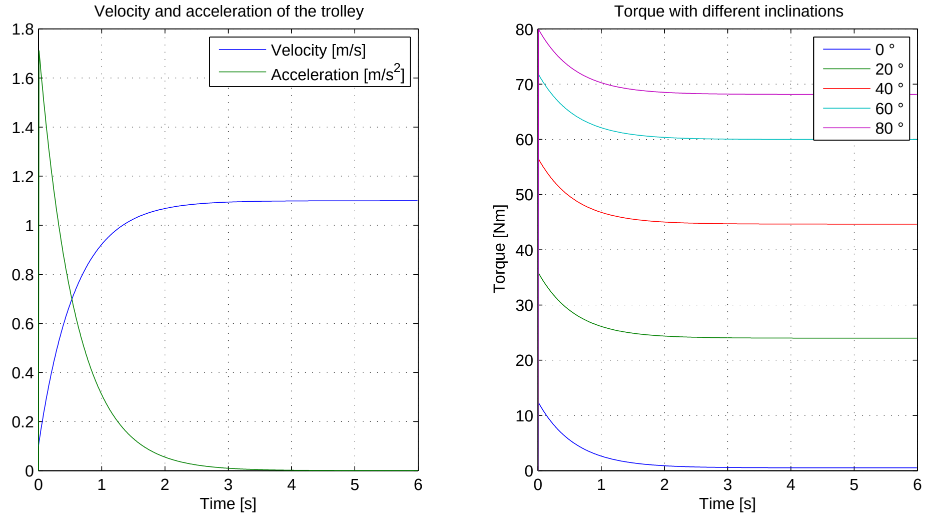


Figure 7.1: The left figure shows the velocity and acceleration used to calculate the torque required in different slopes. The right figure shows the torque required with the velocity and acceleration in the left figure. The torque is calculated for different inclinations and a total mass of 35 kg.

a steady state value depending on the degree of inclination. The values of inclination shown unlikely appears when walking outside, however each step in a staircase behaves as a short slope with a maximum of 90° inclination which is why the high inclination torque values are shown in the graph.

Each torque curve has a high peak in the beginning due to the acceleration. Assume a motor with maximum torque lower than the peak appearing when the inclination is 80°. The only thing affected by cutting the peak is the acceleration at that inclination. At any lower inclination the trolley would behave as expected. Hence the peaks is of little importance when looking at the motor torque required. The maximum motor torque

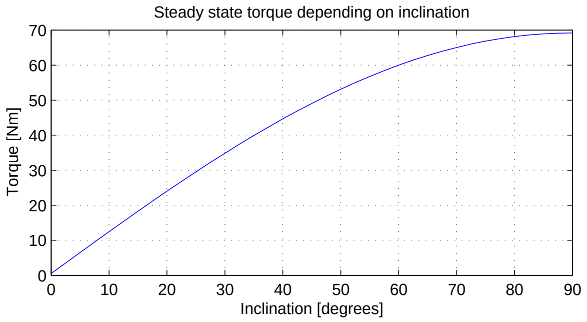


Figure 7.2: Steady state calculations of the total torque as a function of the inclination of the road. In the calculations the total mass of the trolley is assumed to be 35 kg.

required is instead determined by the highest steady state torque. Figure 7.2 shows the steady state torque as a function of the inclination. The maximum torque required is approximately 69.2 Nm at the inclination of 90°. In that case the motor would carry the total mass of the trolley up the stairs since $69.2/r_w = 69.2/0.2 = 346$ N is approximately the same as $m_{tot}g = 343$ N and the difference can be explained by the friction. Regarding slopes outside, they do probably not exceed an inclination of 20° which means 25 Nm is enough to drive the trolley if stair climbing isn't a part of the objective. Also noticeable is that the objective is to give help, not to drive the whole mass of the trolley.

7.1.2 Force measurements

A drive cycle analysis is made using the strain gauge sensors together with the first prototype SGB card as mentioned in Section 6.2.3. The result can be viewed in Figure 7.3. From the results it is deducted that with a load of about 20 kg, a force of approximately

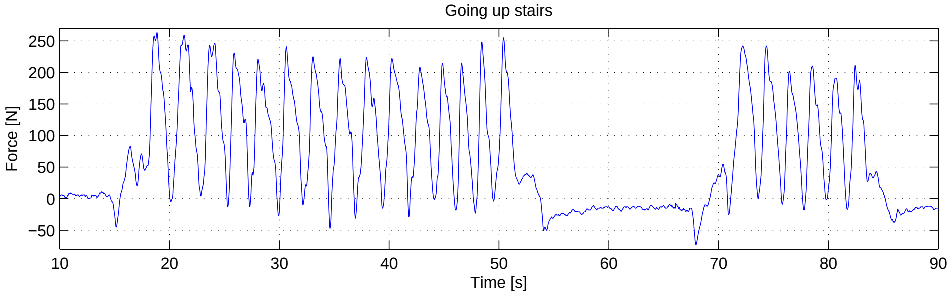


Figure 7.3: Measurements of the force required to pull the trolley up a set of stairs. The results from this test was used in the drive cycle analysis. The total mass of the trolley was 20 kg. Each spike in the graph is one step of the staircase.

250 N is required in order to climb stairs, which is the worst case for the trolley. In Section 2.3 it is stated that the final solution should use big wheels with rough texture to climb stairs. "Big wheels" is here defined as wheels with at least 40 cm diameter, which means a radius higher than one step in a staircase. Together with the knowledge

that the force needed is about 250 N from the results in Figure 7.3 the resulting torque is

$$T = 250 \cdot 0.2 = 50 \quad \text{Nm} \tag{7.4}$$

given from multiplying the needed force with the radius. This means that in order to drive the entire trolley up the stairs, approximately 50 Nm is needed. If compared with the motor torque calculations in Section 7.1.1, where the inclination was set to 90°, this result is lower than expected. This is explained by the trolley weight being different. The torque calculations in Section 7.1.1 is done with a total weight of 35 kg while in this case the total weight is only 20 kg. If the motor torque calculations from Section 7.1.1 would have been remade, the result would correspond to the results given in this section at about 50 Nm.

Now, what about the needed velocity? Normal walking velocity is measured to be approximately 1.4 m/s which is about 1.1 Hz with the current wheel radius at 20 cm. From this it is assumed that if even if you would walk really fast you wouldn't reach above 2 Hz at normal walking velocity. This becomes the second constraint on the motor/motors, they need at least 120 rpm.

7.1.3 Motor

Since there is only one motor available which somewhat fits the drive line specification above, only one motor is used. The motor chosen is a 12/24 V permanent magnet synchronous motor with a maximum torque at 1.1 Nm and maximum velocity 3600 rpm at 12 V or 7200 rpm at 24 V. It has a gearbox attached to it in order to better fulfil the application, with a ratio of 30 : 1. This means that the outgoing axle from the gearbox can give about 33 Nm along with 120 rpm at 12 V or 240 rpm at 24 V. One motor is not enough to drive the entire trolley itself up the stairs as concluded in earlier sections, however it still gives some help.

7.2 Assembly

The mechanical construction is based on an existing shopping trolley. The purpose is to integrate the load cell into the handle and to design a box which is to hold the components such as motor, battery and circuit boards. The box is placed underneath the existing trolley and replaces the original wheels and wheel axle.

7.2.1 Load cell assembly

As mentioned the load cell needs to be integrated in the handle of the trolley so that the force is measured in the right direction. To do this the rod leading up to the handle of the trolley is disassembled and two new parts is built. The two parts attaches the rod again with the load cell in between. In Figure 7.4 the two parts and the load cell is assembled. The load cell is the S-shaped part in the middle of the assembly. The two unused holes in the upper part is connected to the handle while the two holes in the lower part is connected to the lower part of the rod. The finished assembly of the load cell can be viewed in Figure 7.5.

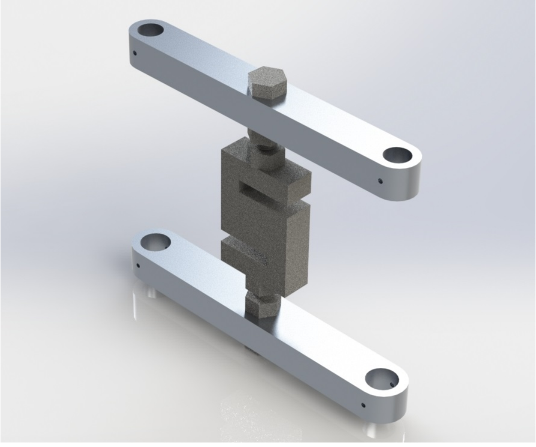


Figure 7.4: Assembly of the load cell. The backside of the trolley is connected to the upper and lower unused holes.

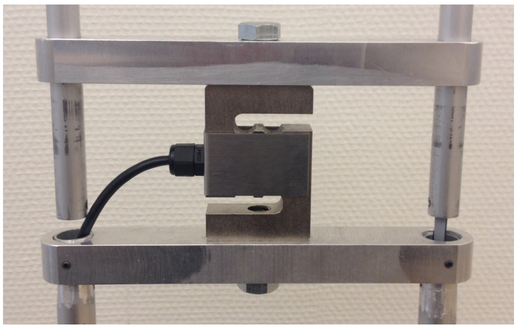


Figure 7.5: Finished assembly of the load cell.

7.2.2 Component box

Figure 7.6 shows the plastic box which holds most of the electrical and mechanical components. The electrical components are the XMC board, the SGB, the motor drive board and a battery. The mechanical components are described below. The design makes it possible to choose between using one or two motors. The finished component box can be viewed in Figure 7.7.

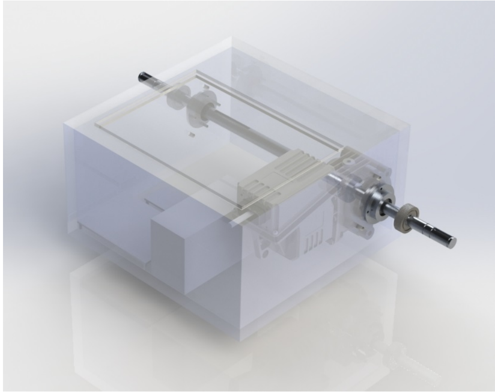


Figure 7.6: Assembly of the component box. The box will be placed under the existing trolley.

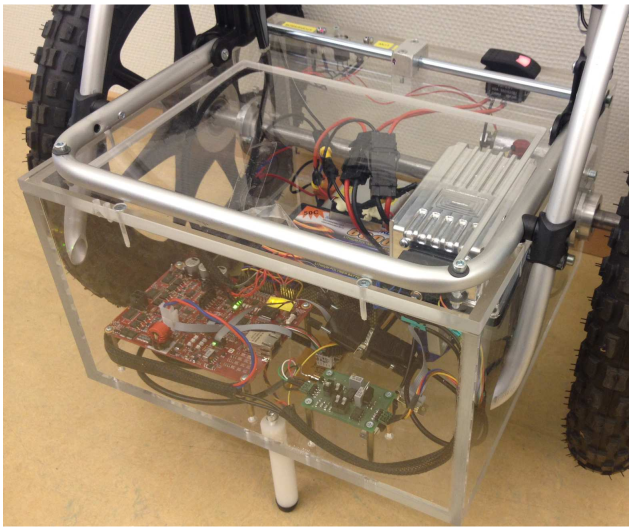


Figure 7.7: Finished assembly of the component box. The box is placed under the existing trolley.

Plastic box

The plastic box is made of acrylic glass which gives the opportunity to see the components while keeping them contained which is preferable on a prototype like this. The box has pre-drilled holes for mounting of the motor and the bearings. The box also has a lid which simplifies assembly and disassembly.

Drive axle

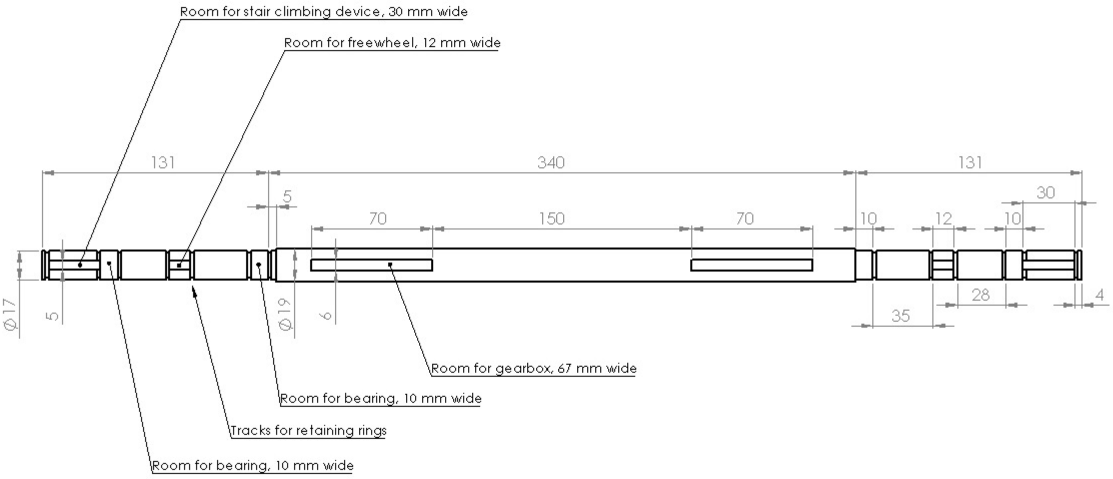

Room for stair climbing device, 30 mm wide
Room for freewheel, 12 mm wide
131
340
131
5
70
150
70
10
12
10
30
Ø17
5
Ø19
6
35
28
4
Room for gearbox, 67 mm wide
Room for bearing, 10 mm wide
Tracks for retaining rings
Room for bearing, 10 mm wide

Figure 7.8: Design of axel.

Even thou two of the specified motors would be needed to give a reasonable amount of help, only one motor was used in the design due to a shortage of the specific motor. This introduces another problem, how does the trolley turn when both wheels are fixed to the same axle? An answer to this is to connect the wheels to the axle with freewheels. Freewheels works as normal bearings when turning in one direction (axle and wheel can rotate independently to each other). However, when turning in the other direction the axle and wheel fixates and they turn together. This makes it possible to drive the wheels forward using the motor, however the user can always pull faster than the motor without having to pull the motor and gearbox as well. This enables the possibility to also turn while walking with the trolley (one wheel can move faster than the other). The motor basically determines the minimum velocity of the trolley but the user can still walk faster. This solution gives demands on the axle. The drive axle is designed considering a number of different demands:

- The motor uses a 6 mm parallel key and a 19 mm axle diameter. Possibility of two motors is wanted.
- Two bearings should attach the axle to the box with room for retaining rings.
- Each wheel is attached to the axle with a freewheel and a bearing with room for retaining rings. The freewheel uses a 5 mm parallel key and a 17 mm axle diameter
- Outside each wheel there needs to be a possibility to attach a mechanical part used for stair climbing.

The final axle design is shown in Figure 7.8.

Gearbox attachment

For the motor and gearbox to be attached to the plastic box a metal distance is needed. The distance has 6 threaded holes. The gearbox is mounted to three of them and the plastic box is mounted to the other three. The thickness is 15 mm so that the motor does not touch the plastic box.

Bearing attachment

The purpose of the bearing attachment is to lock the bearings to the box so that the axle is kept in place. The bearing fits perfectly into a cup which is assembled on the sides of the box.

8 Software Implementation

The software implementation includes the controller structure as well as the communication implementations.

8.1 Controllers

The controller software is implemented in the XMC board as mentioned in Section 6.1. The purpose of the software is to read the user force input and give a control signal to the motor depending on what controller is active (proportional velocity, integral velocity or torque controller). The controller software structure is shown in Figure 8.1. As expected, the program starts with a calibration and initiation sequence which calculates the force offset and sets the DMD board in running mode. After the initiation sequence, three processes starts which are running with different time periods.

The first process handles the communication with the Arduino. Every other time the loop is running, the program will send a variable or receive a variable update. The Arduino is the user interface and therefore needs all the variables which may be changed by the user, i.e. controller mode, controller parameters and filter time constant. Also battery level is indicated. The Arduino then return codes to the XMC, where each code represents incrementing or decrementing the current variable. One of the code bits represents if the OPC is activated or not so that the XMC always knows if to drive the motor or not.

Process number two is the controller. As often as the user specifies, the process loop samples the user force, calculates control signal depending on the active controller and then sets the motor reference. The motor reference is either a velocity or torque value which is sent to the DMD board. The DMD board uses a built in controller to maintain the given reference.

The third process handles controller mode changes from the user or sets error mode if the DMD board demands it. The mode changes internally first and then sends an update to the DMD board. At the end of the process all controller variables are updated.

8.2 Communication

Some communication is needed between the different boards used. The different protocols used is briefly explained below.

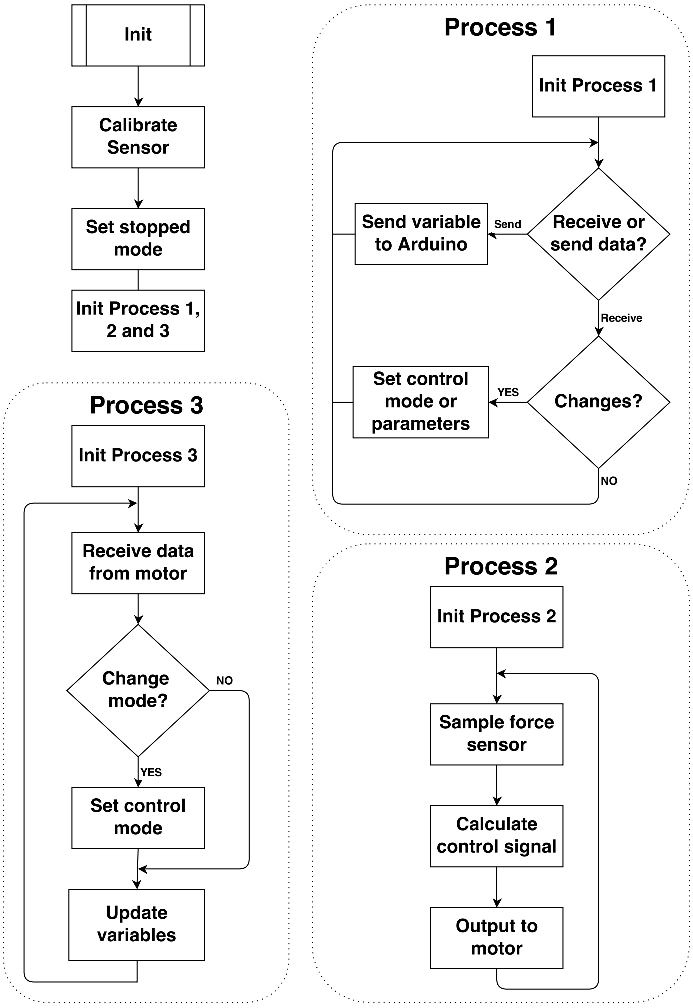


Figure 8.1: Flow chart of the controller and communication software implementation.

8.2.1 CAN

For communication between the main controller board (XMC) and the motor control board (DMD) the CANopen communication protocol is used. This protocol is already implemented in the two boards and is the main reason to why this communication method is chosen. There are also CAN ports available on both boards which also affects the decision.

Establishing the CANopen connection is fairly easy. The XMC board is set to be master and the DMD board to slave. When the XMC tells the motor board to go into a specific operation mode the NMT protocol is used. However, the PDO protocol is used when sending messages about control reference, error codes, voltage levels and so on.

8.2.2 SPI

For communication between the XMC and the Arduino, Serial Peripheral Interface (SPI) is used. The reason to why SPI is chosen in this case is that the Arduino does not have a CAN port and the I2C port is already taken by the LCD shield that is mounted on top of the Arduino. In principal the SPI communication works as follows. The master (XMC in this case) sends a command to the slave (Arduino) and after the Arduino has received the command, the master either sends a variable value or expects the slave to answer with a certain value. Some problems are noted when trying to establish the SPI connection between the XMC and Arduino board, and those are:

- The XMC is speaking SPI at 3.3V, while the Arduino is set to 5V.
- A level shifter implemented to solve the problem in the above bullet point makes the signal distorted which leads to bit losses.
- The cable needed to reach all the way from the XMC to the Arduino is longer than wanted, this could introduce noise in the signal.
- The Arduino board doesn't have built in SPI libraries when in slave mode.

A bi-directional level shifter from Sparkfun is used instead of the original level shifter. This solved the first two bullet points. SPI protocol code is written for the Arduino in order to make it a slave which solves the last bullet point. However, for the third bullet point there isn't much to do in hardware. There are noise introduced which sometimes resulted in problems. For instance, when the Arduino is supposed to answer the XMC with a certain value, it sometimes won't send back the correct information. But instead it sends back the same command it have just received from the XMC. This is easily solved in the software by resending the same message again every time the answer is the same as the command.

9 Results

This chapter contains the results from measurements of the trolley in different environments such as stairs and slopes.

9.1 Friction in gearbox

Below this section measurements is done to evaluate the controllers implemented to the trolley. To make the measurements more accurate the loss of torque due to frictions in the gearbox has to be known. The help which the user experiences does not match the measured helping force since the friction of the gearbox will absorb some of the energy. Figure 9.1 shows the torque needed to rotate the drive axle in a certain constant velocity when the load of the trolley it self is removed. This means only the torque needed to overcome the friction of the gearbox is applied. The torque required is approximately 4 Nm which gives a loss in force of $4/r_w = 4/0.2 = 20$ N from the measured to the experienced motor force where r_w is the wheel radius. This has to be taken into consideration when looking at the following figures.

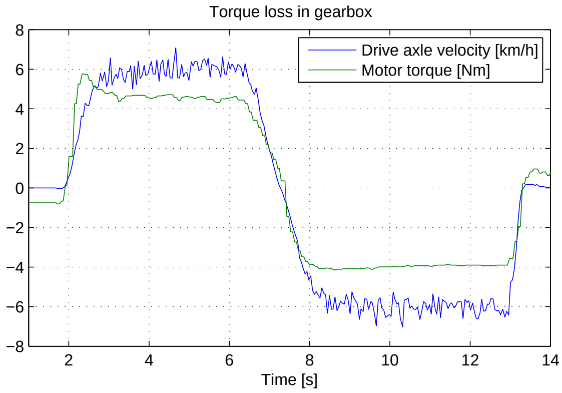


Figure 9.1: Measurements of the torque provided by the motor when the only load is the gearbox friction and the drive axle inertia. When going forwards in the first half of the graph the wheel inertia is also included.

9.2 Controller results

This section is divided into different tests where the velocity controllers and the torque controller are compared in performance. All tests are performed with a load of 15 kg resulting in a total weight of 35 kg. In the figures below, the applied user force (u_{user} from Chapter 3.1 and 4) is presented along with the provided motor torque (measured in the DMD board) scaled to motor force using the wheel radius. When looking at the user force signal it has to be taken into account that some force is required on to hold the trolley in a certain pitch angle. When standing still the required force is measured to 15 N. This means the user force directly adding velocity to the trolley is the rest. The noise like behaviour in the user force signals below is actually force variations which occur in each step the user takes.

9.2.1 Normal walking

The user and motor force applied while walking on flat ground are shown in Figure 9.2. The stopped mode (top graph) shows the user force required to pull the trolley with no help from the motor. For this test the user force needed is approximately 30 N in stopped mode.

Proportional velocity controller

In the proportional velocity controller (second graph from the top) the user has to apply almost the same amount of force as in the stopped mode. The torque provided by the motor is determined to maintain a certain velocity which in this case is low due to the low force needed to move the trolley forwards. Since the trolley uses freewheels, the wheels may rotate faster than the drive axle, which means the motor may not provide any help at all. Only the torque needed to overcome the gearbox friction (calculated above) is added. However, the user force needed in the acceleration phase is lower than in stopped mode which gives a nice feeling when walking with the trolley.

Integral velocity controller

The integral velocity controller (third graph from the top) makes the user force smoother and lowers the average force applied to just over 20 N. However the time for the motor to reach the correct velocity is quite long which means the user won't get any help while accelerating the trolley. When walking normally with no slope, the acceleration phase is where most help is needed.

Another drawback with the integral velocity controller is that it is very important that the trolley is started in walking position, i.e. when the user is holding the trolley with a certain yaw angle. If this isn't done the force needed to keep the trolley in that yaw angle is seen as an error signal which the integral action compensates for. Hence the steady state force is when the trolley pushes on the hand so that the total force on the load cell is zero.

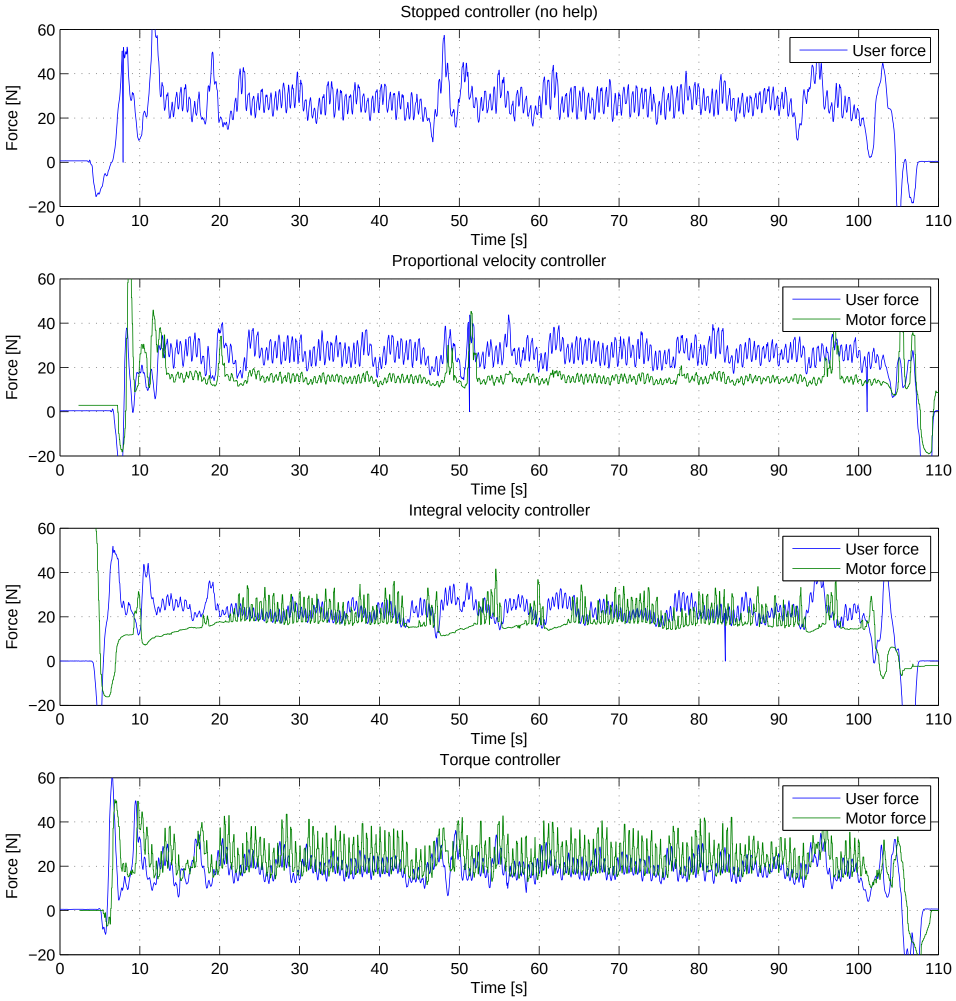


Figure 9.2: Measurements of force applied by user and force provided by motor. The measurements was taken when walking normally indoors with no slope.

Torque controller

While using the torque mode (bottom graph) the average user force is approximately 20 N. The torque is responsive and active. Every small spike in the graph is a step from the user where the user force increases momentarily and the motor torque follows.

9.2.2 Short steep slope

Figure 9.3 shows the user and motor force while walking up a short slope. The inclination of the slope is measured to a maximum of 9.3°. The stopped mode shows the force needed to pull the trolley with no help from the motor. For this test the force needed is approximately 80 N on average.

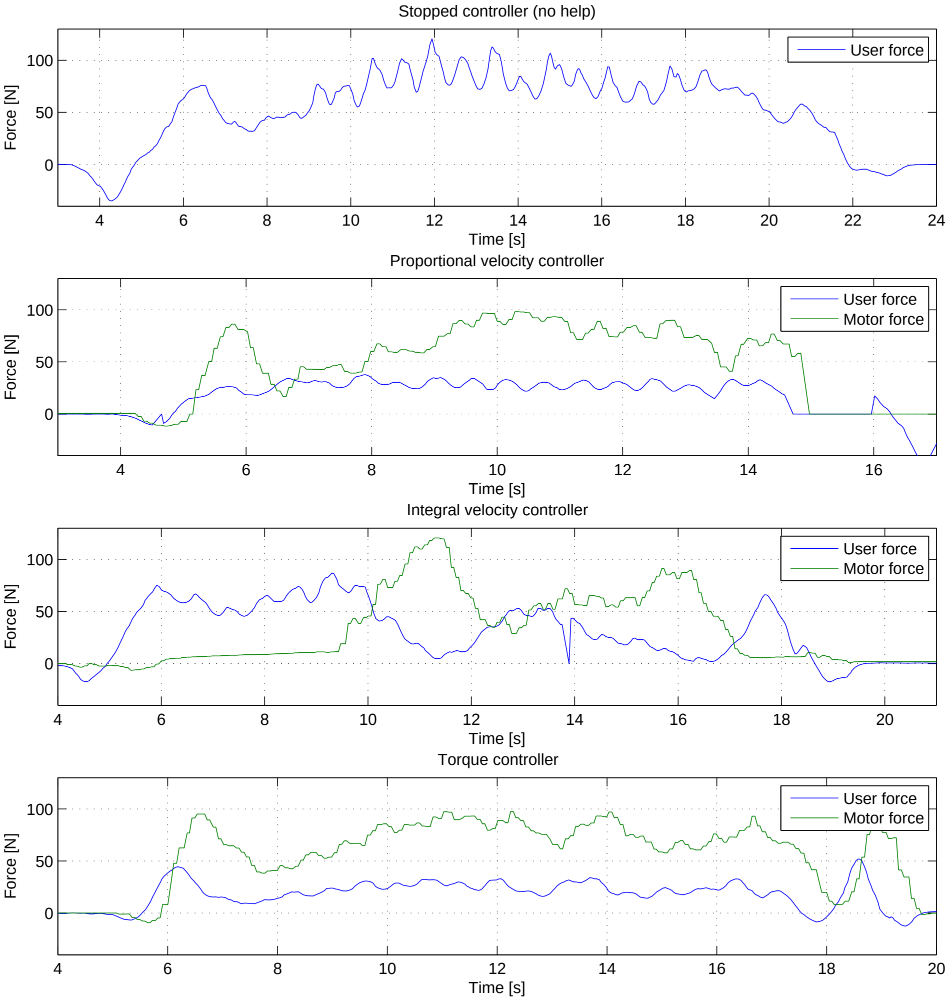


Figure 9.3: Measurements of force applied by user and force provided by motor. The measurements was taken when walking up a short slope with approximately 20 degrees inclination.

Proportional velocity controller

The proportional velocity controller handles the slope smoothly. The applied user force is 30 N which is slightly higher than in the previous test. This means it feels as if one is pulling the trolley on a road with no incline. However, in this case the motor torque needs to be higher for the trolley to move at the same velocity. This is verified in the graph where the motor force is approximately 80 N.

Integral velocity controller

At first the integral velocity controller doesn't give any noticeable force since the angular velocity of the wheels is higher than that of the drive axle. When the angular velocity of the drive axle and wheels are the same, the user force is lowered and the motor force increased. The controller is not optimized to this slope and do not find a stable velocity in such a short distance.

Torque controller

The torque mode handles the steep slope well, the user force is approximately 25 N. In this test the proportional constant in the controller is increased from the previous test. Otherwise more user force would have been needed since the motor force is proportional to the user force as stated in Section 4.2.

9.2.3 Short stairs

Figure 9.4 shows the user and motor force while walking up a short set of stairs. For this test the maximum force required in stopped mode is approximately 350 N which is as carrying the whole mass of the trolley including load.

Proportional velocity controller

The maximum user force needed to pull the trolley up the stairs when using the proportional velocity controller is 280 N. That is probably the best any controller will do since the motor has reached its maximum torque. The controller maximizes the torque when trying to achieve a certain velocity which means maximum help will be given until the trolley has climbed the first step and the reference velocity is lowered.

Integral velocity controller

The integral velocity controller also has the lowest possible user force. However the stair climbing have to be done very slowly so that the trolley doesn't hit the next step in high velocity. This is an unpleasant behaviour. Also it takes some time for the motor to get to the maximum helping force.

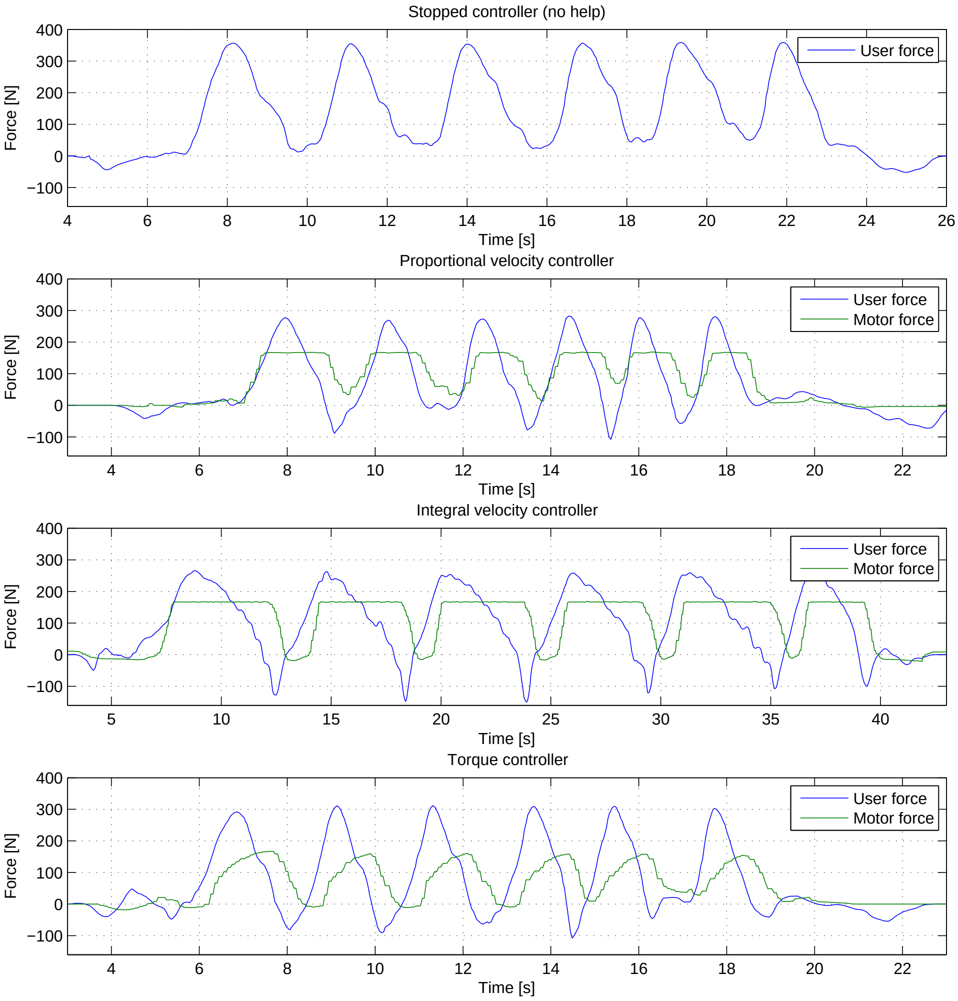


Figure 9.4: Measurements of force applied by user and force provided by motor. The measurements was taken when walking up a short set of stairs.

Torque controller

As previously mentioned the motor force is proportional to the user force in the torque controller. This means the motor force won't reach its maximum value if the proportional constant is to low. However, if the proportional constant is to high the torque derivative might be to aggressive for the motor electronics to handle. Hence, the torque controller doesn't work optimally when walking up stairs.

10 Verification

In this chapter, measurements is compared with simulations and calculations to verify or discard the initial theories. The required motor help, the controller behaviour and the over all requirements and wishes are verified.

10.1 Required help from motor

In Section 7.1.1 the required motor torque is calculated based on the total weight of the trolley and the slope of the road. The worst case is of course driving up stairs where the total torque required is calculated to 70 Nm. The motor used have a maximum torque of $1.1\eta = 33$ Nm where η is the motor gearbox ratio (Section 7.1).

The measurements in Figure 10.1 is taken in order to look closer at what happens in one step of a staircase. The trolley is placed on the edge of the step and held there for a couple of seconds. Unlike the previous stair measurements, this gives a steady state force required to keep the trolley from falling down the stairs when standing as far out on the edge as possible. The measurements are done with the stopped controller and with the proportional velocity controller where the total steady state force required is approximately 225 N in both cases. The total measured force differs from the real total force since the measured total force only notices the component in the direction of the rod to the handle. Hence the relation between the real force F_R and the measured force F_M is

$$F_R \cos\psi = F_M \tag{10.1}$$

where ψ is the pitch angle shown in Figure 3.4. The pitch angle is assumed to be about 45° which makes the real force $F_R = 225/\cos 45 = 318.2 \approx 320$ N. The trolley with load weighs 35 kg. The difference in force could be explained by the trolley not being placed perfectly on the edge.

Based on the real force the torque required is calculated as $T = F_R r_w = 318.2 \cdot 0.2 = 63.6$ Nm. When comparing this to the motor torque calculations in Figure 7.2, 63.6 Nm means an inclination of approximately 65° which could have been the tangent of the edge of one step.

Another comparison to be made is the total force in the short slope. In Figure 7.2 a 9.3° incline gives a torque of 11.6 Nm which means a force of $11.6/0.2 = 58$ N. From Figure 9.3 the force needed at the maximum inclination is between 75 and 110 N. Since the force changes a lot in each step the user takes, it is hard to determine a steady state

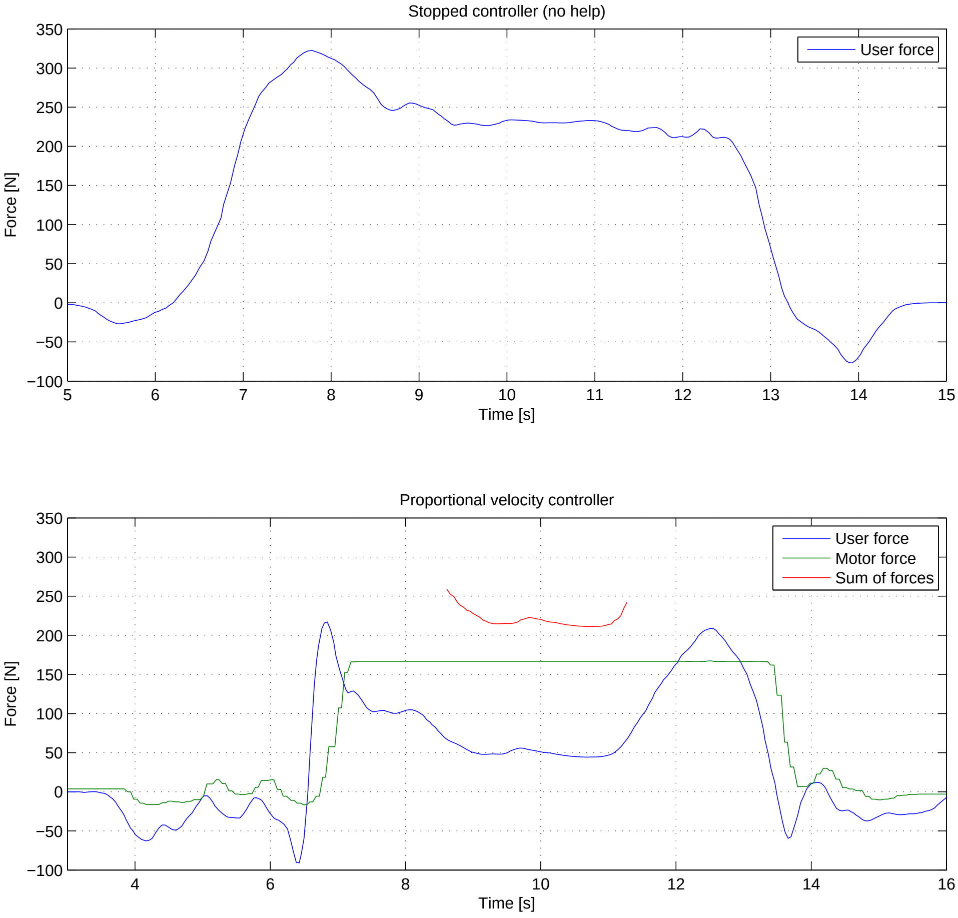


Figure 10.1: Measurements of force applied by user and force provided by motor when standing on the edge of one step to see the total force used to carry the trolley.

value. However the steady state force would most likely be in the lower end of the span. Altogether the motor torque calculations in Section 7.1.1 seems reasonable.

10.2 Controller behaviour

In this section the behaviour of the simulated controllers and the real controllers is compared. In Figure **??** all controller modes are simulated with no inclination on the road.

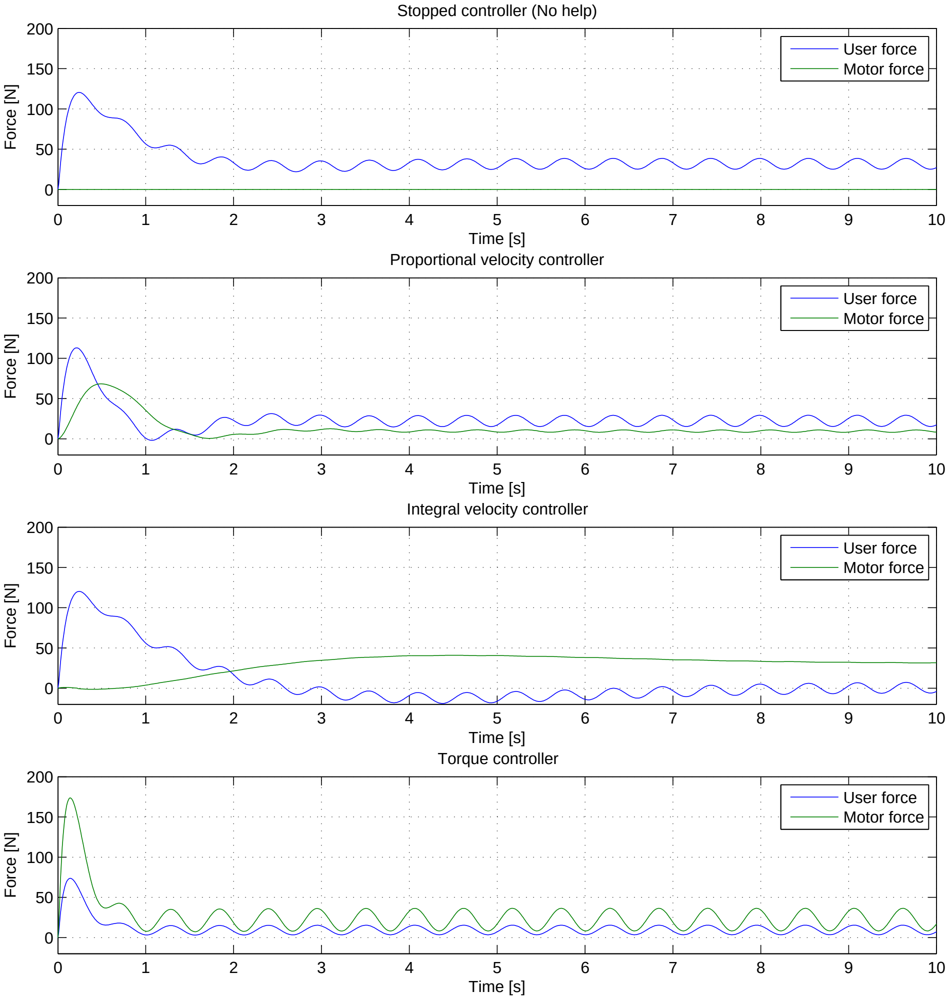


Figure 10.2: Displaying the simulation results for all controllers side by side. Here the force from the user can be compared with the force generated by the motors.

Figure 10.2 is compared to Figure 10.3 where the corresponding measured result is shown. The two figures are in different time frames since the simulated results are step responses while the measured results are whole drive cycles. In stopped mode it is clear that the total force needed is approximately the same in the two figures.

10.2.1 Proportional velocity controller

The proportional velocity controller behaves similarly in the two figures. The user force needed is close to that in the stopped mode due to the low helping force. When the trolley

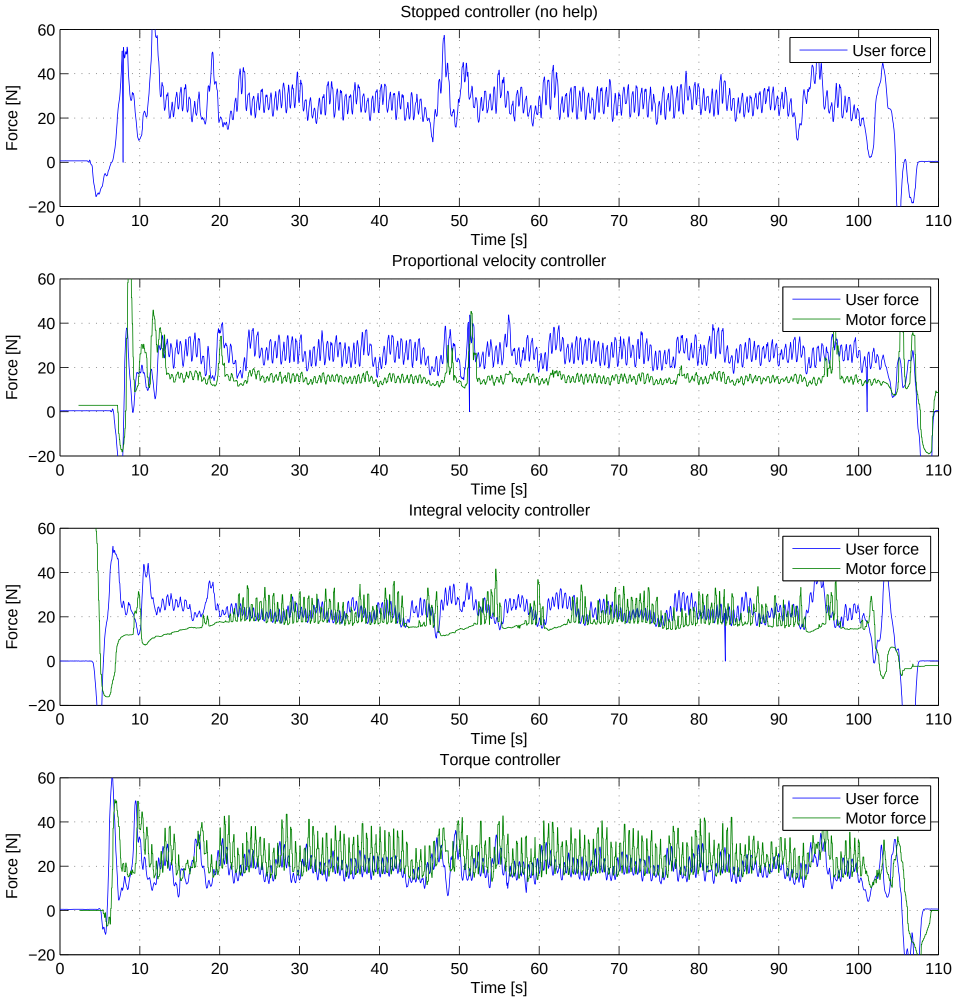


Figure 10.3: Measurements of force applied by user and force provided by motor. The measurements was taken when walking normally indoors with no slope. This figure is the same as Figure 9.2 repeated for easier comparison.

has reached the reference velocity the force will settle at a steady state value. The motor torque then works to overcome the friction. The proportional velocity controller will mostly help the user in the acceleration phase.

10.2.2 Integral velocity controller

When comparing the real and simulated integral velocity controller it is noticed that the simulated controller is more stable. This is expected since the behaviour of a user is hard to predict which means a steady state value is difficult to achieve. Apart from that the behaviour of the controllers are similar. The total user force is lowered but the user doesn't get any help when accelerating the trolley.

10.2.3 Proportional torque controller

The proportional torque controller is very responsive in both simulations and reality. The behaviour of the controller is predictable since the motor strive to give a certain amount of torque, not drive in a certain velocity. It is visible that the motor force follows the user force in both the simulated and real controller.

10.3 Requirements and wishes

In Section 2.1 the initial requirements and wishes for the trolley is described. The final trolley is developed on that basis and fulfils all of the requirements and some of the wishes. Regarding the requirements:

- The ability to give help in different kinds of stairs and slopes is the main goal which is achieved with three different controllers.
- The amount of space occupied by the drive line and electronics is higher than expected. However the user doesn't lose any of the original space since motor and electronics are placed below the bag.
- The trolley handles a lot of weight, however the amount of help from the motor is limited which, past a certain point, makes it heavier to pull the more load it has.
- The drive time only depends on the size of the battery and how many steep slopes and stairs the user faces. The battery is of 6000 mAh and the maximum current is approximately 10 A when going up stairs. The battery lasts longer if the trolley is driven on a road with no inclination with the proportional velocity controller in which case the motor only would give high torque when accelerating.

Most of the wishes is fulfilled in the final solution, however some are not developed to the fullest potential due to lack of time. The wishes that are not fulfilled are:

- Maximize torque at sensorless low speed
- Autonomous drive
- Easy recharging of battery

11 Discussion

The objective is to develop a product based on the shopping trolley which is able to handle rough terrain such as steep hills, stairs and curbstones. The trolley should also be comfortable to pull on flat surfaces. In short the results state that the trolley fulfils the objective. The trolley gives the user help in steep hills and stairs and it is also comfortable to use on flat ground.

The toughest test is off course the stairs where the inclination is the steepest. In that case, the helping force from the motor maximizes and the user has to do the rest of the work. The helping motor force saturates at 170 N which is less than the force needed to carry the trolley without any load. This means that the motor do not actually help with the load when in stairs but only help with most of the trolley weight. It would actually take less force to carry the load without using the trolley. The problem is either that the mass of the trolley itself is too high or the that motor torque is too low. The easiest solution for a prototype like this would be to add a second motor. The mechanics was developed with the opportunity to add a second motor if this result was discovered, however some extra parts would have to be made. Unfortunately the result is discovered to close to the end of the project and there is a shortage of the motor in question.

The mechanical part of the stair climbing works great thou, the grip from the wheels with rough texture is enough to keep it from spinning in the stairs. In Chapter 2.3, final solution, it is stated that the trolley may need an additional mechanical solution in order to climb stairs in addition to the wheels with rough texture. But as it seems, the mechanical solution (which is spring spokes mounted on the sides of the wheels) is unnecessary to solve the objective.

In slopes and when walking on flat ground the result shows that the user gets a reasonable amount of help and that the feeling is nice. The load weight is 15 kg but from measurements the experienced load when walking up a steep slope is 3 kg which is similar to the force indicated when walking on flat ground. This means that the added force from going up a steep hill is eliminated by the helping force.

All in all the concept of the trolley with electric drive is great. It would probably help the elderly a lot when going shopping. With the addition of a second motor, the trolley would feel light even in stairs. One of the drawbacks of the trolley is that it would be expensive in comparison to the non-electric one. Another thing is that the market for the trolley is rather small since it solves such a specific task for a small audience, it would be viewed as a luxury product. However the idea could be transferred to other products such as wheelbarrows, golf bags or carts. In any area where a machine is driven by hand the user force could be measured and used to give an appropriate amount of help accordingly.

12 Conclusion

In this project a smart shopping trolley with electric drive is developed. The trolley measures the force provided by the user and generates an appropriate amount of help based on the measured force. The trolley is able to drive on flat surface, in steep slopes and up stairs. However to reach the full potential in stair climbing a second motor would be needed.

A problem discovered while developing the trolley is that the finished product is heavier than expected and thus makes it hard to operate if the motor isn't running or if it has to be picked up. This is not good since the purpose of the trolley is to simplify for the elderly

For future work the obvious first step would be to add a second motor since that would make the product more useful in stairs and it would be relatively easy to implement. The next step would be to add angle measurements to the controllers to make them smarter. The angle measurements could be used to remove the force offset that occurs due to the pitch of the trolley.

Bibliography

Boström, A. (2013). Rigid body dynamics. Compendium.

Fleming, J. F. (1969). Stair-climbing vehicle. US3450219.

Herbert, Sam, D., A. Drenner, and N. Papanikolopoulos (2008). Loper: A quadruped-hybrid stair climbing robot. Technical report, Department of Computer Science and Engineering, University of Minnesota.

Jacovich, Marissa, L. (2005). Design of a stair-climbing hand truck. Technical report, Department of Mechanical Engineering, Massachusetts Institute of Technology.

Matthies, L., Y. Xiong, R. Hogg, D. Zhu, A. Rankin, and B. Kennedy (1999). A portable, autonomous, urban reconnaissance robot. Technical report, Jet Propulsion Laboratory, California Institute of Technology.

Moghadam, Majid, M. and M. Ahmadi (2007). *Climbing Robots, Bioinspiration and Robotics Walking and Climbing Robots*, Chapter 26. I-Tech Education and Publishing.

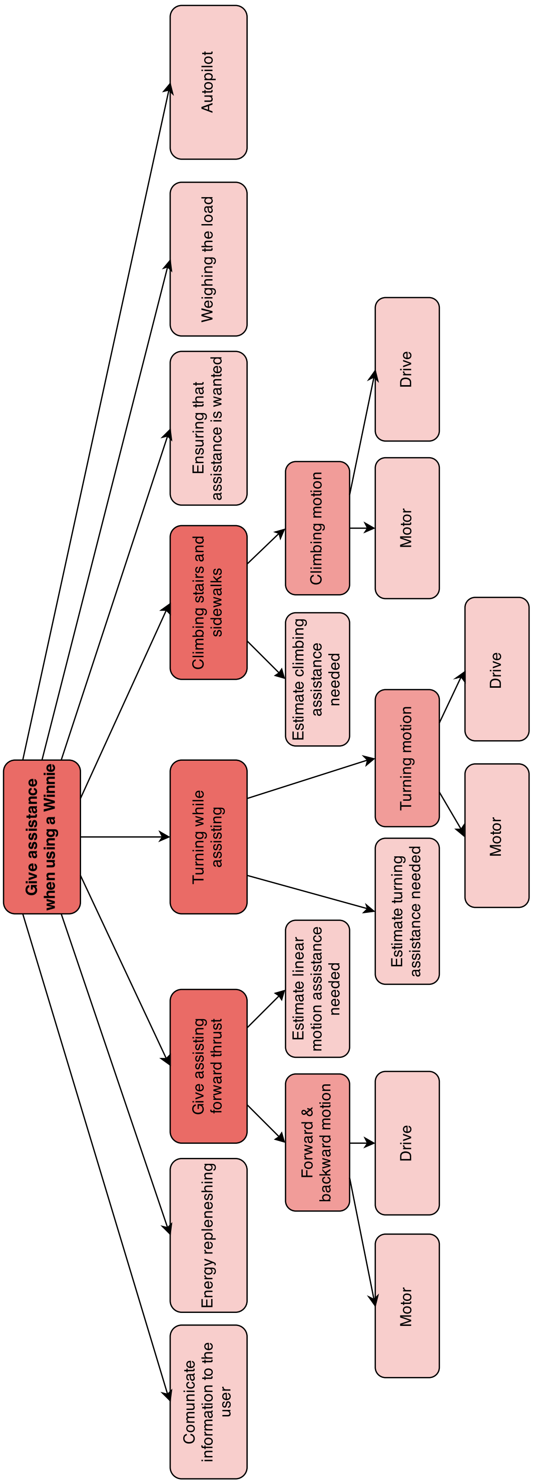
Moore, E., Z. and M. Buehler (2001). Stable stair climbing in a simple hexapod robot. Technical report, Ambulatory Robotic Lab, Centre for Intelligent Machines, McGill University.

Wang, Y., I. Ryojun, S. Hideki, and H. Soichiro (2010). Analysis of human arm characteristics in cooperative motion based on musculoskeletal model. *IEEE RAS AND EMBS International Conference on Biomedical Robotics and Biomechatronics vol. 3*, pp. 301 – 306.

Wu, S. (2013). The electric wheelchair up and down the ladder. [YouTube] `https://www.youtube.com/watch?v=Nfm2Td_NHgk`. (2014-01-27).

Yamamoto, Y. (2009). Mxtway-gs model-based design, control of self-balancing two-wheeled robot built with lego mindstorms nxt. Technical report, Engineering Department, Cybernet Systems CO.

A Function tree


Give assistance when using a Winnie
Autopilot
Weighing the load
Ensuring that assistance is wanted
Climbing stairs and sidewalks
Turning while assisting
Give assisting forward thrust
Energy repleneshing
Comunicate information to the user
Climbing motion
Estimate climbing assistance needed
Drive
Motor
Turning motion
Estimate turning assistance needed
Drive
Motor
Estimate linear motion assistance needed
Forward & backward motion
Drive
Motor

B Partial solutions

Functions		1	2	3	4	5	6	7	8	9	10
Comunicate information to the user		Display	LED lights	Speaker	Haptic feedback	Aromas	None				
Energy repleneshing		Cord	Generative breaking	Inductive charging	Change battery	Diesel generator	Nuclear generator	Dynamo	Biogas generator	Furnace	
Estimate linear motion assistance needed		Strain gauge	Throttle control (like motorcycle)	Preasure plate in handle	Voice command	Angle measurements (trying to stay plain to surface)	Torque/Current induced in motor from pulling the Wagon	User controlled assistance level with buttons	Smartphone app	Transmitter in users pocket (gives distance and direction)	IR/radar/sonar measurements
Forward & Backward motion	**Motor**	BLDC motor	Stepper motor	DC motor	AC motor	Linear motor (rowing motion)	Rockets				
	Drive	Wheels	Caterpillar tracks	Sliding	Stilts (legs)	Airborn					
Estimate turning assistance needed		Strain gauge	Voice command	Torque/Current induced in motor from pulling the Wagon	User controlled assistance level with buttons	Smartphone app	Transmitter in users pocket (gives distance and direction)	IR/radar/sonar measurements	None		
Turning motion	**Motor constillation**	2 motors (one on each side)	1 motor	2 motors (1 motor used sideways)	Several motors						
	Drive algorithm	Set different speed on the motors	Braking left/right side different amount		Human power						
Estimate climbing assistance needed		Throttle control (like motorcycle)	Angle measurements	Voice command	User controlled assistance level with buttons	Smartphone app	IR/radar/sonar measurements	None			
Climbing motion	**Motor requirement**	None	Extra motor	Linear motor (pushing upwards)							
	Drive constillation	Wheels in a triangular constillation	Long caterpillar tracks with rough texture	Extra wheel on rod pushing upwards	Robot arm pulling the Wagon up	Very big wheels with rough texture	Wheels with spokes	Two wheels on each side with spokes that are turnable w.r.t each other	Wheels with spring spokes	None	
Ensuring that assistance is wanted		Pressure plate	Dead mans grip	Use angle measurement	Smart algorithm with existing sensors	ON/OFF button	Smartphone app				
Weighing the load		Scale in the bottom of the bag	Pressure sensors on the wheel axis	Scanning shopping articles	Built in scale in handle (strain gauge)	None					
Autopilot		GPS	Odometer	IR/radar/sonar measurements	Learn route	Kinect	None				

C Total solutions

Total solutions	Comunicate information to the user	Energy repleneshing	Estimate linear motion assistance needed	Forward & Backward motion		Estimate turning assistance needed	Turning motion		Estimate climbing assistance needed	Climbing motion		Ensuring that assistance is wanted	Weighing the load	Autopilot
				Motor	Drive		Motor constillation	Drive algorithm		Motor requirement	Drive constillation			
A	Display	Cord	Angle measurements (trying to stay plain to surface)	AC motor	Wheels	None	None	None	Angle measurements	None	Very big wheels with rough texture	Dead mans grip	Built in scale in handle	None
B	LED lights	Inductive charging	Strain gauge	DC motor	Caterpillar tracks	Strain gauge	2 motors (one on each side)	Set different speed on the motors	IR/radar/sonar measurements	None	Long caterpillar tracks with rough texture	Pressure plate	None	None
C	LED lights	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	None	Very big wheels with rough texture	Smart algorithm with existing sensors	None	None
D	LED lights	Cord	Angle measurements (trying to stay plain to surface)	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	None	Very big wheels with rough texture	ON/OFF button	None	None
E	Display	Generative breaking	Transmitter in users pocket (gives distance and direction)	AC motor	Wheels	Transmitter in users pocket (gives distance and direction)	2 motors (one on each side)	Set different speed on the motors	IR/radar/sonar measurements	None	Two wheels on each side with spokes that are turnable w.r.t each other	Dead mans grip	Scanning shopping articles	Kinect
F	LED lights	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	None	Wheels with spring spokes	Dead mans grip	None	None
G	Display	Cord	Throttle control (like motorcycle)	DC motor	Caterpillar tracks	Strain gauge	1 motor	Braking left/right side different amount	Throttle control (like motorcycle)	None	Long caterpillar tracks with rough texture	ON/OFF button	Scale in the bottom of the bag	Learn route
H	LED lights	Cord	Smartphone app	AC motor	Caterpillar tracks	Smartphone app	2 motors (one on each side)	Set different speed on the motors	Smartphone app	None	Long caterpillar tracks with rough texture	Smartphone app	Scanning shopping articles	Kinect
I	Speaker	Nuclear generator	Voice command	Rockets	Airborn	User controlled assistance level with buttons	2 motors (1 motor used sideways)		Throttle control (like motorcycle)	Extra motor	Long caterpillar tracks with rough texture	Pressure plate	Pressure sensors on the wheel axis	Odometer
J	Display	Cord	Angle measurements (trying to stay plain to surface)	AC motor	Wheels	Strain gauge	1 motor	Human power	None	None	Wheels with spring spokes	Smart algorithm with existing sensors	Built in scale in handle (strain gauge)	None
K	Display	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	IR/radar/sonar measurements	None	Wheels in a triangular constillation	Dead mans grip	None	None
L	LED lights	Cord	Strain gauge	AC motor	Stilts (legs)	Strain gauge	Several motors	Set different speed on the motors	IR/radar/sonar measurements	None	None	Dead mans grip	None	None
M	LED lights	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	Linear motor (pushing upwards)	Extra wheel on rod pushing upwards	Smart algorithm with existing sensors	None	None
N	LED lights	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	None	Wheels with spokes	Dead mans grip	None	None
O	LED lights	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	Linear motor (pushing upwards)	Robot arm pulling the Wagon up	Smart algorithm with existing sensors	None	None
P	Display	Cord	Strain gauge	AC motor	Wheels	Strain gauge	2 motors (one on each side)	Set different speed on the motors	None	None	Very big wheels with rough texture and spring spokes if needed	Dead man's grip	None	None

D Simulink models

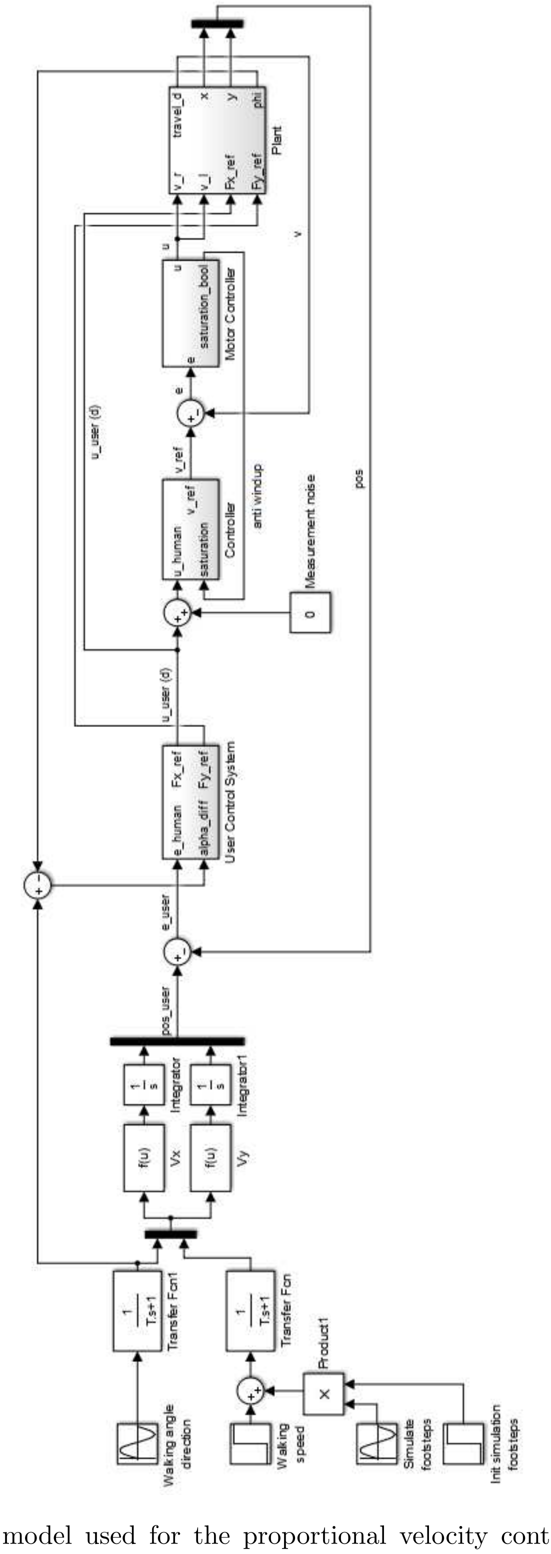


Figure D.1: Simulink model used for the proportional velocity controller and integral velocity controller.

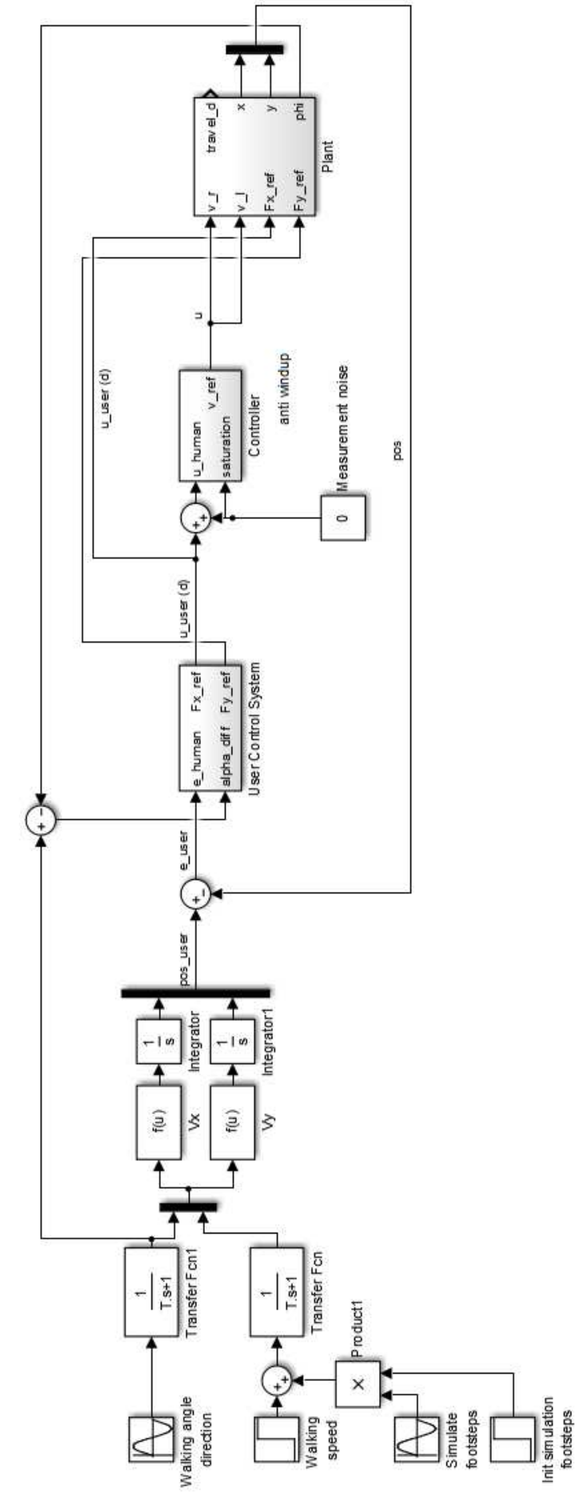


Figure D.2: Simulink model used for the torque controller.

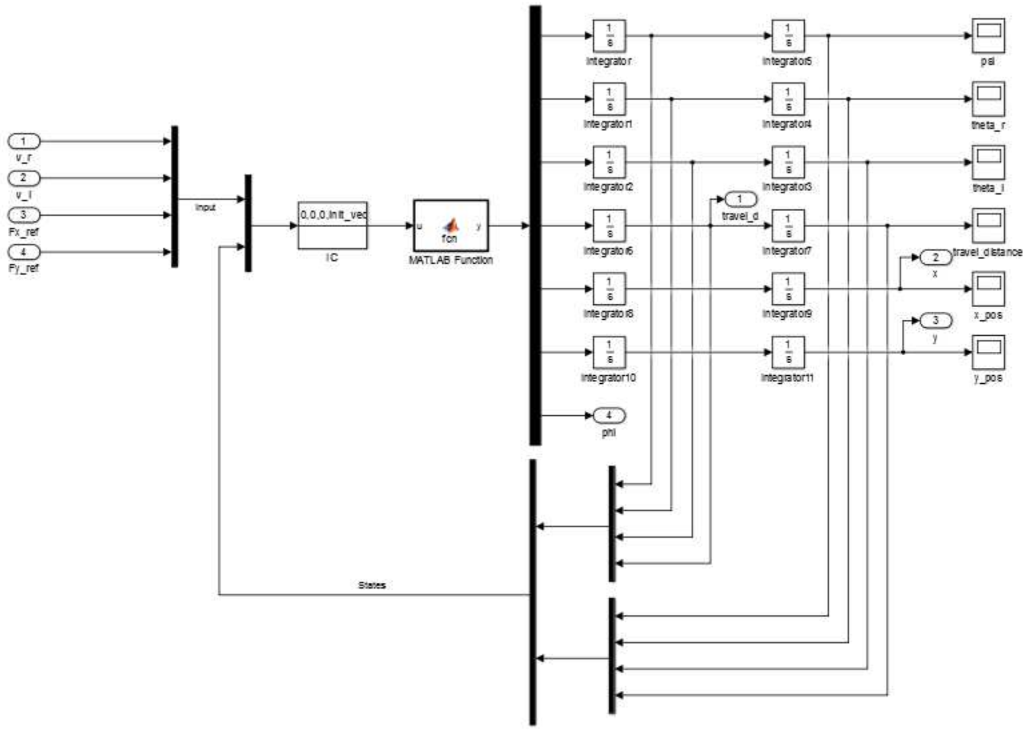


Figure D.3: Simulink model used for the plant.

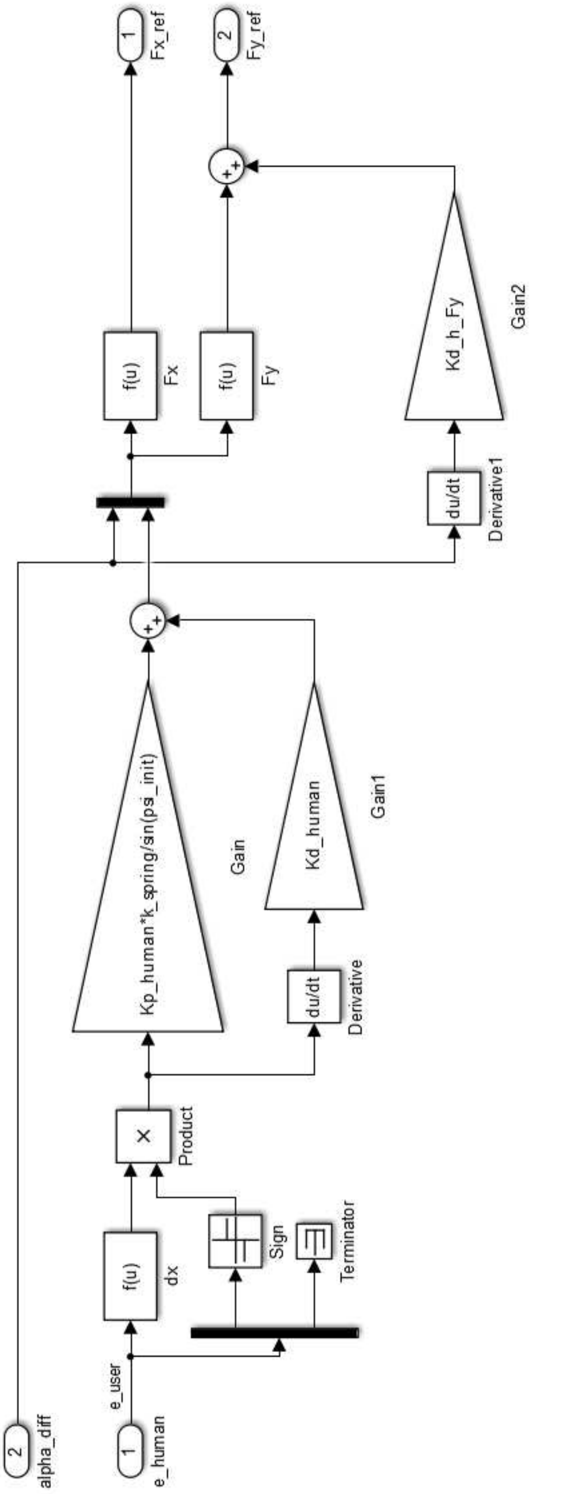


Figure D.4: Simulink model used for the user control system.

E PCB

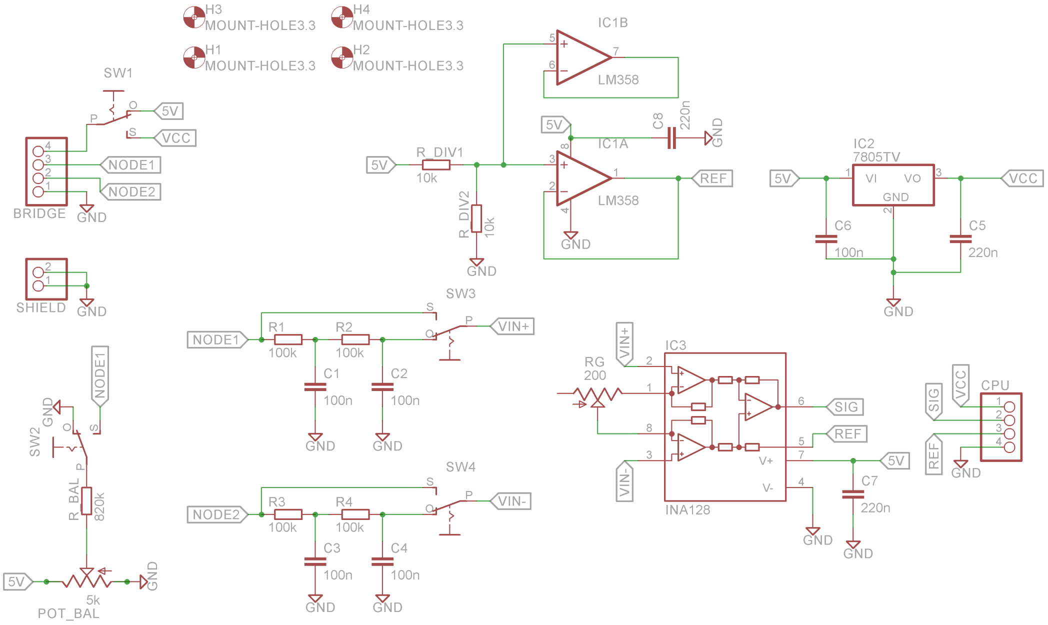


Figure E.1: Noise measurement of the different force measurement configurations.

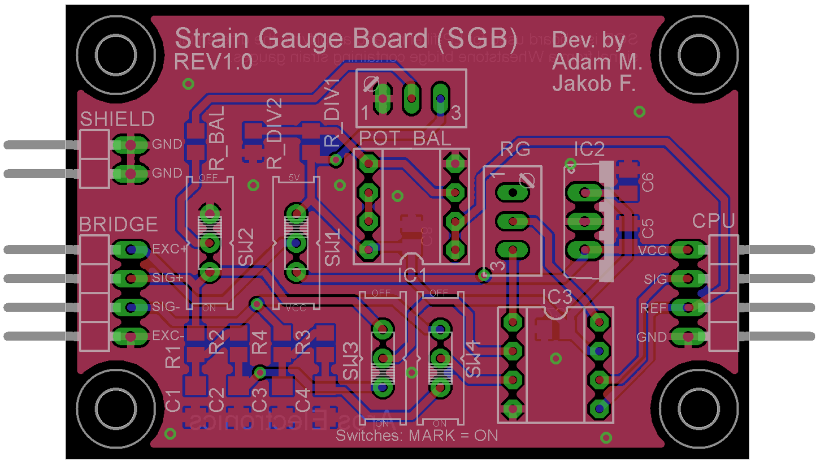


Figure E.2: Noise measurement of the different force measurement configurations.