ArtiTrax II

Martin Karlsson

Luleå University of Technology
Dept. of Computer Science and Electrical Engineering

12th January 2011
ArtiTrax is an articulated wheelchair with four-wheel drive. It was constructed in 2006 by nine students in a mechatronics project. The purpose was to create a wheelchair with improved off-road capabilities. By the end of that project, a person could control the wheelchair, either from the driver’s seat, or remotely with an external joystick using the Bluetooth protocol. However, the ArtiTrax control algorithm was never completed and had a few issues. This master thesis is about converting the ArtiTrax wheelchair into a small scale research platform for articulated steering. To this purpose, an onboard PC is mounted, speed measuring encoders are attached to the motors, a wireless router ensures remote control capabilities, and the software is prepared to transfer data to and from an external computer. The vehicle uses CANbus as the main means of inter-vehicle communication. The software relies on the Generic Intelligent Machine network infrastructure (GIMnet). With a person in the drivers seat, ArtiTrax II behaves like an ordinary wheelchair by responding to the onboard joystick commands. When an external computer is connected to the ArtiTrax network, a custom controller can be implemented with the MATLAB® software. The controller can be evaluated in realtime, with or without a driver, and is easy to modify as it is located on an external computer. No modifications has to be done on the ArtiTrax II platform. The external computer can connect with WLAN and be located in the wheelchairs immediate vicinity, or on the other side of the globe by using the Internet.
My work on the ArtiTrax II research platform marks the end of my five year studies in electrical engineering at Luleå University of Technology. This project has without doubt been my greatest challenge so far. Although, at the same time, it has been both fun and captivating. Sometimes I have been experimenting with custom control algorithms, just for the fun of it.

I want to thank my supervisor Håkan Fredriksson for all his support and good ideas, Volvo Construction Equipment for the funding of new hardware to ArtiTrax II, Prof. Kalevi Hyyppä who suggested this project as a suitable master thesis, and Ralf Forsberg at Permobil for the software modifications to the power amplifiers and answers to many questions.

Martin Karlsson
# CONTENTS

## Chapter 1 – Introduction
1.1 Background ............................................. 1
1.2 Purpose .................................................. 2
1.3 Goal ..................................................... 2
1.4 Limitations .............................................. 3
1.5 Outcome ................................................ 3

## Chapter 2 – Basic Platform
2.1 CANbus .................................................. 6
2.2 Motors ................................................... 7
2.3 Power Amplifiers ....................................... 7
2.4 Control Panel .......................................... 8
2.5 Power Supply .......................................... 9

## Chapter 3 – Hardware Modifications
3.1 Joint Motor ............................................. 11
3.2 Encoders ............................................... 14
3.3 Power Amplifiers ....................................... 14
3.4 Angle Decoder .......................................... 15
3.5 Inertial Measurement Units ......................... 16
3.6 CANbus 1 and 2 ....................................... 16
3.7 Onboard PC ............................................ 18
3.8 Wireless Router ....................................... 18
3.9 Emergency Stop and Remote Control Switch .... 18

## Chapter 4 – Software
4.1 GIMnet .................................................. 21
4.2 Arch Linux ............................................. 21
4.3 Open-Source Software Tools ....................... 22
4.4 GIMnet Layout and Hardware Interaction ....... 22

## Chapter 5 – Results
5.1 Local Control Algorithm ......................... 25
5.2 Sampling Rate ......................................... 29
## 5.3 PID Controllers

### 5.3.1 Wheel Speed PID Controller

### 5.3.2 Joint Angle PID Controller

### 5.3.3 Joint Angle Speed PID Controller

## 5.4 Discussion

## 5.5 Future Work

### Appendix A – ArtiTrax II CANbus messages

#### A.1 CANbus 1

#### A.2 CANbus 2
CHAPTER 1

Introduction

1.1 Background

Figure 1.1: ArtiTrax I.

Figure 1.1: ArtiTrax I.
The ArtiTrax wheelchair is the result of a mechatronics project conducted by nine students at Luleå University of Technology in 2006 [1]. The purpose was to create a wheelchair with improved off-road capabilities. The base was created from the rear part of two Permobil Trax wheelchairs with a joint in between for articulated steering. This joint can also tilt the front with respect to the rear to ensure that all wheels are on the ground even in rough terrain. Each wheel is powered by a DC motor thus making ArtiTrax four-wheel driven. A fifth DC motor can, via a chain transmission, steer the joint in either direction, although its abilities to do so was never confirmed. The vehicle is equipped with a CANbus to transfer data between modules such as joystick, PC and motor control units.

The control algorithm for the articulated wheelchair was never fully implemented. However it was still possible to drive ArtiTrax at slow speeds. This was achieved by having the onboard PC to monitor the CANbus for joystick messages and calculate new speed messages to the motor control units. This control algorithm did not use the joint motor.

1.2 Purpose

The purpose of this master thesis is to upgrade the ArtiTrax wheelchair by modifying both hardware and software, so that it can be used as a platform in research projects as well as educational projects.

1.3 Goal

The ArtiTrax wheelchair should be driveable directly from the onboard joystick. Custom control algorithms should be easy to implement and remote controlling done by wireless TCP/IP communication. The hardware requirements are:

- CANbus is the main means of communication. One exception is remote control were wireless TCP/IP is required.
- Speed measuring encoders should be attached to each wheel. The solution must keep the current width of the vehicle.
- Gyros are to be mounted, one in the front- and one in the rear part of the vehicle.
- Evaluate the joint angle decoder.
- Evaluate the joint angle motor.
- Replace the PC/104 with UNO-2172 PC.
- Evaluate if the existing joystick can be used for remote control.
1.4 Limitations

- Evaluate if the existing display can be used in the new system.

Software requirements:

- ArtiTrax should be controllable by joystick and from MATLAB®.
- The GIMnet infrastructure is to be used.
- Logging of sensor data should be possible.
- The software in the motor controlling units should be updated by Permobil, so the motor input voltage can be controlled directly.

1.4 Limitations

- The focus is to prepare the research platform, not developing a sophisticated regulator to control it. However, it should be possible to drive and steer the wheelchair in a satisfying way.

- As the wheelchair turns to the right or left, there is a risk of something getting stuck at the joint. There are two safety switches that are toggled if something is pushed against them. They will not be implemented since they are not critical for the project.

- In the rear of the ArtiTrax wheelchair, there is an ultrasonic range-finding unit. This unit is connected to the CANbus, but in this project, no software drivers are prepared for it. This is because it has no direct use on an articulated research platform.

1.5 Outcome

The final ArtiTrax II platform, seen in Figure 1.3, can be controlled in two modes: the local and the remote control mode. The local mode is the default way to control the wheelchair. It is ready to be used directly after the onboard PC has finished booting. Tilting the joystick to the left or right causes the joint angle, see Figure 1.2, to change, and tilting the joystick front or back causes the wheelchair to move forward or backwards.

To use the remote mode, an external computer should connect to the same network as the ArtiTrax PC. With the use of a socket client, prepared for the MATLAB® software, the remote computer can passively monitor all the vehicles' measurement data. It is possible to send drive commands to the motor control units. When a remote drive command is received, the wheelchair changes to the remote mode and will stop the local controller. As an example, if the command to set the voltages for each motor to a positive value was
sent remotely, the wheelchair will drive forward regardless of how the joystick is pointing. If no new remote commands are sent within 500ms of the last received command, the wheelchair returns to the local mode. It is also possible to have MATLAB® read the joystick directions and send drive commands that makes the wheelchair movement dependant on the joystick. This implies that it is easy to implement and evaluate custom control algorithms.

Figure 1.2: Articulated vehicle with the joint angle $\alpha$ defined.

Figure 1.3: ArtiTrax II.
This chapter describes the ArtiTrax hardware prior to this thesis. The ultrasonic range finding unit, remote control Bluetooth module and onboard PC are omitted. The angle decoder is reserved for the next chapter.

Figure 2.1: Hardware in the rear of ArtiTrax I.
2.1 CANbus

ArtiTrax uses a controller area network bus (CANbus) for inter-vehicle communication. A CANbus is a two-wire serial network running through a number of modules, each capable of receiving and transmitting messages. Since it is a broadcast bus, a message sent, is received by all modules, even if they do not need it. Each CAN message consists of an identifier, followed by a data length code and the data. The identifier is also the priority of the message. If two modules start sending at the same time, the message with the lowest priority will have to wait [2]. One of the benefits of using a CANbus is that modules can be added or removed anywhere along the line without interfering with other modules on the bus. The CANbus on ArtiTrax operates at 250kbit/s and uses the CAN standard 2.0A [1], which has a 11 bit identifier. The bus is also extended with an additional key-line and ”dead man switch”-line (DMS-line). The key-line controls the power amplifiers on and off state. The DMS-line is a safety feature that will prevent the wheelchair from moving when the joystick is in its neutral position.

Figure 2.2: Hardware in the front of ArtiTrax I.
2.2 Motors

Each wheel is attached to a 1:18 gearbox that is powered by a 500W 24V DC motor. These motors has an electromagnetic parking brake that is locked when they are unpowered. There is also an additional DC motor at the joint that can change the joint angle. This motor is a Maxon F2260, rated 80W at 15V and is attached to a 19:1 planetary gearbox and finally a 2:1 chain transmission. After a few practical tests it was apparent that this solution was not a very good one. With the breaks released, the motor could barely change the joint angle, even though the motor was overpowered with 24V.

2.3 Power Amplifiers

All motors, brakes and lights are controlled by motor control units, also called power amplifiers or amps. These units are programmed by Permobil and a new program can be loaded from a computer with a serial port. Each unit is capable of controlling two motors. Since there are five motors on the wheelchair, three amps are required. Amp1 powers the front wheel motors, amp2 the rear wheel motors and amp3 the joint controlling motor. Amp2 is the black box located behind the rear motors in Figure 2.1. These units use pulse width modulation (PWM) at a frequency of 20Khz and has their own wheel speed regulator. When speed commands are sent over the CANbus, the amplifier will set the motor voltages with its internal controller.

There are two requirements for powering the motors, first is that the DMS-line is set, second that speed commands are sent periodically. If the sent speed commands are zero or if the DMS is in the low state, the amplifiers will stop the motors and lock the parking brakes. If the requirements are meet, the power amplifiers will release the brakes and

Table 2.1: ArtiTrax I CANbus, d-sub pinout.
power the motors. Periodically these units report the current drain and motor voltage. Amp1 and 2 are each connected to a relay box and will switch on the lights or sound the horn from CANbus messages.

2.4 Control Panel

The ArtiTrax wheelchair is controlled by a person sitting in the drivers seat, with the right hand on the control panel, shown in Figure 2.3. The driver will get information from the display unit and controls the wheelchair with the joystick and surrounding buttons.

![ArtiTrax I control panel and joystick coordinate system](image)

*Figure 2.3: ArtiTrax I control panel, and to the right, the joysticks coordinate system.*

**Joystick**

The joystick module consists of a joystick, a number of buttons and LEDs. The module is closely linked with amp1 because on an ordinary Trax wheelchair, there are only one onboard joystick and one power amplifier. When the on button is pressed, the module sends a number of messages that amp1 receives. They are configuration parameters that changes the behaviour of the power amplifier before it starts. These parameters can
be customised from the joystick module according to the Mini-Config manual [3]. The unit also displays error codes, produced by either itself or the amp. The errors are also explained in the manual. To change the configuration parameters for the other power amplifiers, the messages from the joystick has to be forwarded by changing their CAN-IDs.

The joystick module periodically sends the front-rear and left-right steering data separated in two messages on the CANbus, as explained in Figure 2.3. Both the key-line and DMS-line status are controlled from this unit. The key-line is set by the on and off button, although it requires that the key (which is in the shape of a 1/4” jack) is inserted. The DMS-line status is in the low state when the joystick is in its neutral position. Pointing it in any direction, will set the line to its high state. As a redundancy, the DMS-state is also periodically sent as a CANbus message. If these two differ, a DMS error is produced and the power amplifiers will power off.

Display

The display module is an independent part of the wheelchair and does not send any data, it only listen to the CANbus traffic. By default, it displays the joint angle and a vehicle speed estimation. The two buttons control the menu system and the module can also display the current drained by each motor and the distances to any rear obstacles. Since this is a passive unit it requires that the onboard PC requests measurement data from the angle decoder and ultrasonic range finding unit.

2.5 Power Supply

ArtiTrax has two 12V lead acid batteries connected in series for a 24V rechargeable power supply. Amp1 supplies 24V to all low power CAN modules via pin 9 on the CANbus.
In this chapter, the new hardware and the modifications to the old is described. The remote control Bluetooth module is removed since all remote controlling should be done with wireless TCP/IP.

3.1 Joint Motor

![Joint Motor Image](image)

*Figure 3.1: ArtiTrax II joint motor and a speed measuring encoder (in silver) connected via a belt transmission.*
Since the joint motor and its transmission proved to be too weak, a new solution had to be produced. Practical tests on ArtiTrax have showed that a good angular velocity for articulated steering is $20^\circ/s$ which is 3.33rpm for the joint. If the joint angle is moving at this velocity, the corresponding motor velocity can be calculated and the transmissions performance evaluated. The first step is to determine the motor velocity of the ArtiTrax I joint solution. The subscript $i$ for a variable $x_i$ denotes an input and $o$ for $x_o$ an output.

The rotational velocity in and out of a transmission has the relation

$$g = \frac{n_i}{n_o} = \frac{t_o}{t_i}, \quad (3.1)$$

where $n_i$ and $n_o$ is the in and out rotational velocities in rpm, $t_i$ and $t_o$ the number of teeth and $g$ the gear ratio [4]. The input $M_i$ and output $M_o$ torque for an ideal transmission is determined by

$$g = \frac{M_o}{M_i}, \quad (3.2)$$

[4]. The requested joint motor velocity $n_i$ on ArtiTrax I can be calculated using

$$n_i = g_t \cdot n_o, \quad (3.3)$$

where the total gear ratio $g_t$ of the planetary gearbox $g_g$ and chain transmission $g_c$ has to be calculated as

$$g_t = g_g \cdot g_c = \frac{19}{1} \cdot \frac{2}{1} = 38. \quad (3.4)$$

Then, the motor velocity can be calculated by

$$n_i = g_t \cdot n_o = 38 \cdot 3.33 \approx 127. \quad (3.5)$$

According to the F2260 datasheet [5], the motor has a no load velocity of close to 4000rpm. By using a gearbox with a higher reduction ratio, the required torque from the motor is reduced and it will run at a higher velocity to reach $20^\circ/s$ at the joint. The gearbox with the highest reduction ratio that fit is Maxon GP62A with the gear ratio 236:1. By using this gearbox the new motor velocity

$$n_{i\text{ new}} = g_{t\text{ new}} \cdot n_o = (\frac{236}{1} \cdot \frac{2}{1}) \cdot 3.33 \approx 1570, \quad (3.6)$$

is still lower than the no load velocity. However, the motor torque is much reduced. By using Equation 3.2, the torque delivered by the two transmissions can be compared as a ratio when they have the same motor torque input $M_i$.

$$\frac{M_{o\text{ new}}}{M_o} = \frac{g_{t\text{ new}}}{g_t} \cdot \frac{M_i}{M_i} = \frac{236 \cdot 2}{38} \approx 12.4. \quad (3.7)$$
The result show that the new transmission ideally requires less than 12 times the torque compared to the previous solution.

To further improve the control of the joint, the motor is replaced with a Maxon RE-50, rated 200W at 24V [5]. However, this motor turned out to have some properties that needed to be solved. When it was powered by amp3, the motor temperature increased even though it received a 50% PWM duty cycle input, which should result in 0V DC for the motor. This is because the motors electrical time constant is so low that a significant current can run through it every half PWM period. The time constant can be calculated from the equation,

$$\tau = \frac{L}{R}, \quad (3.8)$$

by using the motors resistance and inductance. By adding an inductor with a ferrite core in series with the motor, the time constant is increased and the ripple current suppressed. The added inductor has a value of 7.9mH and will increase the time constant from 0.83mS to 71mS. This is simulated in Figure 3.3, from the schematic in Figure 3.2, as the two cases, with and without the inductor. It is shown that the added inductor removes most of the current through the motor. The motor will respond a little slower to a change in DC voltage although it is still fast enough and it will not get heated from an idling input.

![Figure 3.2: Simulation schematic of the RE-50 motor with and without the low pass filtering inductor.](image)
3.2 Encoders

The Hengstler AC-58 CANopen is an absolute encoder with a 12bit singleturn and 12bit multturn resolution [6]. One revolution has 4096 data points and the encoders captures the absolute position over 4096 revolutions. It communicates with the CANopen interface. To be able to measure the speed and implementing PID-controllers for wheel speed, joint angle and joint angle speed, every motor has an encoder connected to its motor shaft. For the wheel motors this is done via a flexible coupling, see Figure 3.4, and for the joint motor, two gears and a drive belt, see Figure 3.1. The speed is measured as the difference between two absolute positions 5ms apart. The encoders are configured so that the position is read by a single CANopen sync message and all encoder respond with the requested data. To read the speed, a certain data request message has to be sent to each encoder. Refer to Appendix A for CAN message details.

3.3 Power Amplifiers

For this project, software modifications are done to the power amplifiers. Because of the new encoders, a more accurate control algorithm for motor and wheel speeds can be implemented. Instead of receiving speed messages and estimating the corresponding motor voltage, the amps receive the voltage directly and set it to the motors. This means that it is no longer the power amplifier that controls the motor speed, but a controller in the
onboard PC. At all times, the PC has to ensure that the voltage values sent to the power amplifiers are ramped. If the step change between two sent voltage values are to high, an error is produced and the amp power down. The DMS redundancy is also disabled so the power amplifiers ignores the status of the DMS CAN message. This is required for the remote control switch described in Section 3.9. It is worth to mention that there is still a controller with safety features present in the power amplifiers, and this means that the voltage values sent on the CANbus is not necessarily what the motors receive.

Amp3 that powers the joint controlling motor, need its own modifications. Mainly because the built in safety features in the power amplifiers are based on the 500W wheel motor. When tested, the result was an oscillating PWM output. By using a software from Permobil, custom parameters could be loaded from a computers serial port and tested on the joint motor. When appropriate values was found and the amp could control the motor in a satisfying way, the parameters was sent to Permobil and they prepared the new software for amp3.

3.4 Angle Decoder

At the joint between the front and rear part of ArtiTrax there is a potentiometer attached. The angle of the potentiometer is measured by a microcontroller connected to the CANbus. To read the current angle, a message is sent on the CANbus and the microcontroller responds with a message containing the angle data [1]. Because the reported angle from the potentiometer is an unscaled value, it need to be converted to get the joint angle. Changing the joint angle from max right to max left corresponds to the data values of 52 and 945. This is a straight line relationship, using this relation the angle can be calculated from an ADC measurement with the equation

\[ \alpha = -0.101z + 50.2, \]  

(3.9)
where \( z \) is the unscaled value and \( \alpha \) the joint angle.

\section{Inertial Measurement Units}

For an articulated vehicle it is very useful to detect when the wheels are slipping. This cannot be achieved just by measuring the wheel speed. It does not say anything about how the wheels move with respect to the surface. An inertial measurement unit (IMU) can measure accelerations and rotational velocities. By adding two of these on the ArtiTrax wheelchair, one in the front part and one in the rear part, the orientation and movements of the wheelchair can be estimated. These units give not only new possibilities for research on articulated steering, but also future projects. It was decided to purchase units without the CAN interface and implement that ability with the use of a microcontroller. Analog Devices ADIS16362 is an inertial measurement unit with six degrees of freedom, containing a triaxis digital gyroscope and triaxis digital accelerometer [7]. Olimex AVRCAN is a development board using an Atmel AT90CAN128 microcontroller capable of CAN communication [8]. Both the IMU and development board can communicate with the SPI bus and together they form an IMU with CANbus interface. There were not time to create and implement this IMU unit within this master thesis but the components were obtained and will be completed in a future project.

\section{CANbus 1 and 2}

It was intended to put all hardware on the same CANbus. However, it turned out that the power amplifiers was not compatible with the CANopen standard that the encoders use, so the encoders were instead placed on a separate bus.

One requirement for a functional CANbus is that there are termination resistors at the beginning and end of the bus. These resistor has a value of 120\( \Omega \) and should be connected between the CAN-H and CAN-L pins. To find out where these resistors are located on the ArtiTrax I bus, the resistance over CAN-H to CAN-L on each module is measured. The result show that the only modules with termination resistors are amp1 and amp3.

The previous bus layout is not optimal since amp1 is located in one end and amp3 is located in the middle of the bus. The new layout for the first bus has amp1 and amp3 placed in each end, see Figure 3.5.

For the second bus, the termination resistors are enabled for the encoders in each end, see Figure 3.6. The first bus pins is kept the same as the previous bus, see table 2.1. For the second encoder bus, there is no use of the DMS-line or key-line so they were omitted. The voltage supply is also reduced to 12V from a 50W single output DC-DC converter. See table 3.1 for CANbus2 pinout.
3.6. CANbus 1 and 2

<table>
<thead>
<tr>
<th>pin</th>
<th>function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
</tr>
<tr>
<td>2</td>
<td>CAN-L</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
</tr>
<tr>
<td>5</td>
<td>NC</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
</tr>
<tr>
<td>7</td>
<td>CAN-H</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
</tr>
<tr>
<td>9</td>
<td>+12V</td>
</tr>
</tbody>
</table>

*Table 3.1: ArtiTrax II CANbus2, d-sub pinout.*

*Figure 3.5: ArtiTrax II CANbus 1 layout.*

*Figure 3.6: ArtiTrax II CANbus 2 layout.*
3.7 Onboard PC

The new PC is an Advantech UNO-2172-C22E [9]. It has a 1.5Ghz Celeron processor, 1Gb ram, two Ethernet adapters and four serial ports. A solid-state hard drive is chosen since ordinary spinning disk hard drives are susceptible to vibration and shocks. The computer need to communicate on two separate CAN-buses. By using two CAN232 adapters [10], connected to the computers serial ports, the communication act just like ordinary RS-232. The adapter ensures that messages are sent and received properly according to the CAN standard. The transfer rate is limited to the serial ports 115 200 baudrate.

3.8 Wireless Router

The router is a Linksys WRT54GL [11], and its main purpose is to provide the wheelchair with its own local area network. The onboard PC is connected to the router by an Ethernet cable. Other computers can connect to the same local area network by either a cable or wireless Ethernet. The original firmware of the router is replaced by dd-wrt [12]. It is an open-source software that provides additional functionalities which could be useful in future projects.

3.9 Emergency Stop and Remote Control Switch

One condition for powering the motors is that the motor control units receive a high signal on the DMS-line. The DMS-line is controlled by the joystick module and is set high every time the joystick is moved off centre. For this reason there is a problem when the
vehicle is supposed to be remote controlled. If the computer is sending speed messages to the power amplifiers and the joystick is at the center, the wheelchair will not be moving regardless of the amp input. As soon as the joystick is moved off centre, the DMS-line is set high and the motors will respond to input commands. This problem was first rectified by the use of a remote control switch that was set every time ArtiTrax was going to be remote controlled. This switch was connected between +24V and the DMS-line via a 15kΩ resistor, thus ensuring a high state when the switch is conducting. However, there were disadvantages of this solution. If the switch was left in its on-state when the power amplifiers started, it would produce a DMS-error and the amps had to be restarted with the switch turned off. If the switch was left in its on-state, the power amplifiers would not shutdown from the joystick off button. It was also easy to forget this switch, since it had to be switched on for every remote control session, and switched off before every shutdown, to avoid problems.

There is a way to combine the remote control switch with an emergency stop button to suit both needs. The emergency stop is a feature that should prevent the wheelchair from moving when activated (pressed down). If the emergency button connects the DMS-line to the ground when it is pressed, and sets it to a high state when it is released, the vehicle can be remote controlled when button is released and not drivable at all if it is pressed down.

![Safety switch schematic with RLoad as the pull-up resistor in the joystick unit.](image)

*Figure 3.8: Safety switch schematic with RLoad as the pull-up resistor in the joystick unit.*

Normally the DMS-line is set high by a pull-up resistor in the joystick module. To get the emergency button to have full control over the DMS-line, this resistor need to be
Figure 3.9: Parametric sweep simulation of the emergency stop button. \( R_{\text{Load}} = 1\, \text{k}, 5\, \text{k}, 10\, \text{k}, 15\, \text{k}, 20\, \text{k} \). After 1ms the emergency button is activated and grounds the DMS-line.

grounded when the button is pressed down. This is done by the transistor in Figure 3.8. Since the value of the pull-up resistor \( R_{\text{Load}} \) in the joystick module is unknown, a few different values are simulated to ensure a proper operation. The results show that the emergency stop button will be fully functional for all of the simulated \( R_{\text{Load}} \) values, see Figure 3.9.

Figure 3.10: ArtiTrax II and its router, computer, main power switch, fuses and 12V DC-DC converter mounted above the batteries.
This chapter is about the software on ArtiTrax II. All programs are located on the onboard PC except for the socket client which lies on an external computer within the same network.

4.1 GIMnet

GIMnet stands for generic intelligent machine network and is a communication infrastructure especially designed for robotics. The network consists of at least one tcpHub and a number of modules connected to it [13] [14]. The modules are ordinary programs using the GIMnet Application Programming Interface. When connected to the hub, they can send and receive messages to/from other modules. The tcpHub and modules can be running on a single computer or be spread out over the network by means of tcp/ip communication. Each message contain a major and minor identifier. The major id describes what type of message it is, for example, a message containing a time measurement. The minor id is a specification, for example, if there are several time measurements from different modules, the minor id can specify a time message from module, 1, 2, 3, etc. A GIMnet module can also provide a service for a type of message. This means that other modules can subscribe to this service and receive these messages continuously as new are available. At Luleå University of Technology, GIMnet has been successfully implemented on the MICA wheelchair [15].

4.2 Arch Linux

The GIMnet tcpHub has to run on a Linux platform and there are a variety of Linux distributions to choose from. Arch Linux is lightweight Linux distribution [16], meaning that after the installation there is a minimal operating system without any graphical user interface. It is up to the user to install the required packages to get the desired features.
The main benefits over using a fully-fledged solution are ensuring fast boot times and a small consumption of the systems resources.

4.3 Open-Source Software Tools

As the onboard PC is not intended to have any display or input devices connected, all modifications has to be done with an external computer over the network. For this reason the onboard PC is running the OpenSSH server [17]. This allows a user to log in with a SSH client and execute commands. With use of the X.Org implementation of the X11 windows system [18], it is possible to run X11 clients on the PC which forwards their graphics over SSH to the X11 server on the remote computer. This means that it is easy to modify the ArtiTrax software by using a SSH compatible PC and start a remote X11 session over the SSH connection. To use GIMnet, the source files, as well as the specific ArtiTrax code, has to be compiled. This is done using the GNU Compiler Collection [19].

4.4 GIMnet Layout and Hardware Interaction

As one of the fundaments of GIMnet is modularity, the ArtiTrax II software is split to a number of modules, see Figure 4.1. Each module is assigned a particular task.

- **gimiCAN1** initializes serial port 1 and contains the driver for the CAN232 unit. All messages on CANbus 1 are provided as a GIMnet Generic CAN service with minor id 0. Generic CAN messages that are sent from GIMnet to this module are forwarded to the CANbus.

- **gimiCAN2** initializes serial port 2 and contains the driver for the CAN232 unit. All messages on CANbus 2 are provided as a GIMnet Generic CAN service with minor id 1. Generic CAN messages that are sent from GIMnet to this module are forwarded to the CANbus.

- **gimiWheel** is the main controller for the wheels and continuously sends voltage messages to two of the power amplifiers on CANbus 1. Depending on the input, the module either ramps the output to a voltage or uses a PID-controller to set the desired wheel speed. Necessary wheel speed measurements are subscribed from gimiCAN2.

- **gimiJoint** controls the joint motor by continuously sending voltage messages to the power amplifier on CANbus 1. The module can control the motor in three ways, either by setting the motor voltage, joint angle or joint angle velocity. This is achieved by using two separate PID-controllers and a ramp function. The module subscribes to all Generic CAN services for the necessary measurement data.
• **gimiMaster** determines the behaviour of ArtiTrax by toggling between local and remote control. By default the module monitors the joystick and responds to its movement by sending drive commands to gimiWheel and gimiJoint. When remote drive commands are received from gimiSocketServer, the module switches to the remote control mode and forwards these drive commands to gimiWheel and gimiJoint. If no new commands are received within 500ms it will return to local control mode. The module contain drivers that request measurement data from encoders and angle decoder. It subscribes to all Generic CAN services for measurement data.

• **gimiSocketServer** forwards messages in both directions as soon as a socket client connects. By subscribing to all Generic CAN services, it will first process and convert the measurement data before sending it to the MATLAB\textsuperscript{R} client. This socket server is a modification of the one that was developed for the MICA wheelchair in a master thesis [15].

• **The socket client** is a java object that is loaded by a remote computer with the MATLAB\textsuperscript{R} software. The object is running in the background as a separate thread and holds a number of variable pairs. Each pair contain data associated with a particular type of measurement and its time stamp. It is continuously updated as soon as new data arrive from the socket server. By accessing the methods within the java object, the variables can be read, commands can be sent and data logging performed. To create a control algorithm, the written MATLAB\textsuperscript{R} program should contain a loop that continuously reads new measurement data, such as the joystick directions, and processes them into motor commands, which are sent back to the wheelchair. This socket client is also, like the socket server, a modified version from the MICA wheelchair [15].
Figure 4.1: GIMnet layout and its connections to the hardware.
This chapter gives details about the control algorithm that is implemented to drive the wheelchair from the joystick. It does also discuss the various results and give suggestions for improving the ArtiTrax II research platform.

5.1 Local Control Algorithm

To develop a control algorithm for an articulated vehicle, it is important to understand how the individual wheels rotate as the vehicle moves. If there is no slip between the wheels and the surface, the vehicle motion can be summarised as follows:

- If the vehicle is driving forward or backwards, with a constant joint angle that is not equal to zero: The inner and outer wheels will follow two different circles, with the outer rotating faster than the inner. This is shown in Figure 5.1.

- If the vehicle is standing still and the joint angle is increasing (right turn): The two wheels on the right hand side will approach each other and the ones on the left will move further away, as shown in Figure 5.2. This is the same for a decreasing joint angle but the wheels rotate in the opposite directions.

- If the vehicle is driving forward, or backwards, and at the same time the joint angle is changed: The rotation of the wheels is the sum of the two previous cases. This means that all four wheels will propel the vehicle along two constantly changing circles.
Figure 5.1: An articulated vehicle driving forward at a constant joint angle. The outer wheels are going faster than the inner wheels.

Figure 5.2: Changing the joint angle on an articulated vehicle, causes two wheels to approach each other and the other two moves away.

When ArtiTrax is controlled from the joystick in the local mode, the wheelchair uses an open-loop controller that computes the voltage values to the motors from the joystick inputs. This algorithm is based on the two cases, separated into the joysticks x- and y-axis. Pointing along the y-axis should result in a movement along the two circles in Figure 5.1, and pointing along the x-axis should change the joint angle as in Figure 5.2.
The joystick produces two inputs (defined in Figure 2.3)

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} = \begin{cases}
-100 \leq X \leq 100 \\
-100 \leq Y \leq 100
\end{cases}
\]

(5.1)

The inputs are scaled to appropriate values to avoid an over-sensitive control. The scaling can be modified to make the wheelchair go faster or slower, but also give it a more or less responsive steering.

\[
\begin{bmatrix}
J\text{dir} \\
J\text{spd}
\end{bmatrix} = \begin{bmatrix}
X/3 \\
Y/2
\end{bmatrix}
\]

(5.2)

Finally, the resulting wheel motor voltages can be calculated by adding the values together according to the control algorithm

\[
\begin{bmatrix}
U_{FrL} \\
U_{FrR} \\
U_{ReL} \\
U_{ReR}
\end{bmatrix} = \begin{cases}
J\text{spd} + J\text{dir} & J\text{spd} \geq 0 \ (Forward) \\
J\text{spd} - J\text{dir} & 0.9 \cdot (J\text{spd} - J\text{dir}) \\
0.9 \cdot (J\text{spd} + J\text{dir}) & J\text{spd} < 0 \ (Reverse)
\end{cases}
\]

(5.3)

There is a distinction between driving forward and reversing. This is to make sure that the wheelchair always get more power to the front wheels which may result in a less unstable drive experience. This control algorithm does not have any compensation for the joint angle. By only pointing the joystick along the y-axis (speed), all wheel motors are getting the same voltage value, for all joint angles. There is no obvious disadvantage of this as long as the grip between the wheels and ground are good. The inner wheels will be forced to slow down in the same way as the outer wheels are forced to speed up.

In Figure 5.3 it is shown that pointing the joystick along the y-axis causes the wheelchair to drive forward or backwards. In 5.4 the vehicle is standing still but the joint angle is changing. This is caused by the joystick x-axis movement, and the two wheels on the front and rear always rotate in opposite directions. The combination of driving forward and changing the joint angle becomes the positive speed displacement seen in Figure 5.5. The reason to the jagged wheel speeds seen in these three examples, are the human factor, ArtiTrax II is driven by a person from the drivers seat.
Figure 5.3: ArtiTrax II driving forward and backwards using the local controller. The joystick points along the y-axis.

Figure 5.4: ArtiTrax II standing still and steering to the left and right using the local controller. The joystick points along the x-axis.
Figure 5.5: ArtiTrax II driving forward and turning to the left at the same time using the local controller. The joystick points to a combination of the x- and y-axis.

When driving on a slippery surface such as snow, it becomes hard to change the joint angle because the wheels slip. Driving by moving the joystick along the y-axis can result in an unwanted change in the joint angle. By extending the control algorithm to include the joint motor, this problem is solved. The high gear ratio between the joint and joint motor makes sure that the motor has to be powered for the joint angle to change. The input to the joint motor is

\[ U_{\text{joint}} = J_{\text{dir}}, \]  \hspace{1cm} (5.4)

and can be scaled to make the steering more or less sensitive.

### 5.2 Sampling Rate

The encoders and angle decoder have no default sampling rate. Because the data has to be requested from both units before they transmit, the sampling rate can be controlled. The sampling rate hardware limitation for the angle decoder is about 100Hz. This is because it takes 10ms for the analog to digital conversion to complete \[1\]. To have a good safety margin for the angle decoder, 50Hz is more suitable. Although when this sampling rate was tested, the result was an overloaded CAN232 unit on bus2. The sampling rate had to be reduced to 30Hz to get reliable and periodical measurements.
from the encoders. When the wheels are in the air, they rotate with a maximum speed of 160rpm. This would result in a speed of 3.73 m/s for the wheelchair on the ground. At this speed, there is at least one sample every 12 cm, but in reality, the wheelchair can never reach such a speed. It will be lower because of the loads on the motors. During normal driving and manoeuvring it is more likely about one sample every 3 cm. A decreased sampling rate has the benefit of making the loads on the CANbus smaller. There is less risk of two modules start sending at the same time which causes a transmission delay for one of them. Reducing sampling rate can actually result in a lower time delay for the measurements.

5.3 PID Controllers

For the open-loop control algorithm, it is sufficient to only be able to set motor voltages. This is not always the case and advanced controllers might require the wheels rotate to at a certain speed and have the joint angle to stay at a specified value. This requires closed-loop controllers. A proportional-integral-derivative (PID) controller is chosen because it is both common and usable to a wide range of systems. The controller tries to keep the error between the setpoint and actual point as small as possible. The output from the controller to the system has the equation.

\[ o_c(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}, \]  

(5.5)

with the error

\[ e(t) = r(t) - o(t), \]  

(5.6)

where \( r \) is the setpoint and \( o \) the actual point [20]. For all three PID-controllers in Ar- tiTrax II, the parameters can be changed at any time from a client in MATLAB®. The examples that follow will show the default controllers, although they are not necessarily high performing.

5.3.1 Wheel Speed PID Controller

Figure 5.6 show a P-controller for the wheel speed. It has a large steady-state error, and the overshoot is caused by the voltage controller remaining in the power amplifiers. Figure 5.7 is the same controller with an added integral part. It removes much of the error, although the high acceleration causes the wheel top slip on the surface. Adding a differential parameter to this controller can counteract the acceleration. Figure 5.8 shows this with an exaggerated differential gain.
5.3. PID Controllers

Figure 5.6: Wheel speed P-controller. Setpoint = 80 RPM and $P = 1.5$.

Figure 5.7: Default wheel speed PI-controller. Setpoint = 80 RPM, $P = 1.5$ and $I = 0.07$.

Figure 5.8: Wheel speed PID-controller. Setpoint=80 RPM, $P = 1.5$, $I = 0.07$ and $D = 2$. 
5.3.2 Joint Angle PID Controller

The joint angle P-controller in Figure 5.9 does almost reach the set value. In Figure 5.10, the added integral part makes sure the setpoint is reached. This controller performs well for any angles. In Figure 5.11, a movement between $35^\circ$ and $-40^\circ$ is done. It reaches a maximum speed of over $40^\circ$/s which is more than sufficient for a fast and responsive articulated steering.

Figure 5.9: Joint angle P-controller. Setpoint = $40^\circ$ and $P = 3$. 
5.3. PID Controllers

Figure 5.10: Default joint angle PI-controller. Setpoint $= 40^\circ$, $P = 3$ and $I = 0.04$.

Figure 5.11: Default joint angle PI-controller. Setpoint $= -40^\circ$, $P = 3$ and $I = 0.04$. 
5.3.3 Joint Angle Speed PID Controller

A proportional controller is first tried to set the wanted joint angle speed. It results in a steady-state error, see Figure 5.12. There are some oscillations to the motor speed but the joint angle follow a fairly straight line. In figure 5.13 the added integral part reduces the steady-state error although the oscillations remain. Adding a differential part suppresses the large oscillations and the joint angle line is even straighter.

![Figure 5.12: Joint angle speed P-controller. Setpoint = 20°/s, P = 10.](image)
5.3. PID Controllers

Figure 5.13: Joint angle speed PI-controller. Setpoint = 20°/s, P = 10 and I = 0.2.

Figure 5.14: Default joint angle speed PID-controller. Setpoint = 20°/s, P = 10, I = 0.3 and D = 1.
5.4 Discussion

Artitrax II works very well as a research platform for articulated steering. Both the wheel and joint motors are powerful enough for most situations. Although, the joint motors ability to turn the wheelchair at a high friction surface, such as asphalt, without the help of the wheels, is not determined. If the parking brakes are on, it is very likely that the brakes will slip, before the wheels start to glide, or the joint motor is unable to turn the joint.

The local control algorithm performs well and as an open-loop controller it is not prone to instability. With an experienced driver, it is possible to drive both fast and accurately. One of the benefits of the control algorithm is that it is not dependant on the joint motor. It just behaves a bit differently when the motor is or is not connected. Sometimes control algorithm development might require that the joint motor is disconnected. This is done by removing the chain between the joint motor gearbox and the joint. Because of the local modes duality, it can always be used to control the wheelchair.

The advantage of the ArtiTrax II platform is that it use the MATLAB® software, which is an excellent environment for control algorithm development. When this kind of work is performed, it is done from a remote computer. Although, if MATLAB® is installed in the onboard computer, custom controllers can run locally.

In case of an emergency such as a malfunctioning control algorithm, the best way to stop the wheelchair is by pressing the emergency stop button. Its location makes it easy to reach for the driver, see Figure 3.7. When the button is pressed the motor voltages drop to zero. Because of this, the wheelchair will behave in two ways. If the motor voltage is large enough, the drop to zero volt will cause an out of limit step change. As explained in section 3.3, these are not allowed. The power amplifiers will produce an error and the brakes will lock. Afterwards the power amplifiers need to be restarted. If the motor voltage is within the limit step change, the wheelchair will roll to a halt and then lock the brakes. The important thing is that in both cases the wheelchair will come to a complete stop. If the wheelchair is remote controlled and there is not a person in the drivers seat, it might be hard or impossible to reach and press the emergency stop. The best way in this situation is to terminate the running MATLAB® code. Since the wheelchair is not receiving any new remote control commands, it will timeout after 500ms and switch to the local mode. Because of the neutral position of the joystick, the local mode will not power the motors and the wheelchair will roll to a stop. If the wheelchair is moving fast at the time of termination, it will have time to move a significant distance before it halts.
5.5 Future Work

When inertial measurement units and other modules are to be added on the ArtiTrax II platform, they should neither be added to the first nor the second CANbus as long as the CAN232 devices are used. Because the load on the buses are quite high, adding these modules will probably require the sampling rate to be reduced even further. The main limitation is the serial port and its maximum 115 200 baudrate. Preparations have been made on the second CANbus by increasing the bus speed to 500kbit/s. By changing the CAN232 controller to a high speed CAN controller, the sampling rate can be increased and more modules supported.

A more accurate motor control can be achieved by replacing the power amplifiers, since it is not known how the internal voltage control algorithm is implemented and affects the performance. If these units are replaced, the onboard computer need to be configured to respond with a certain CANbus amp1 OK message after the joystick has sent the amp1 configuration parameters, see Appendix A. Otherwise, an error is produced by the joystick module. It is beneficial, especially for safety, if the replacing motor control units are customised to take advantage of the existing DMS-line and parking brakes.

When there is a driver in the seat, the ArtiTrax wheelchair has a high centre of gravity, and there is a risk of the wheelchair to roll over when manoeuvring in slopes. The centre of gravity can be lowered by placing the driver outside of the wheelchair, and this can be achieved by having a GIMnet module, on an external computer with connected joystick, to override the onboard joystick. This way, driving will be more safe and the wheelchair can handle a more demanding terrain.
APPENDIX A

ArtiTrax II CANbus messages

A.1 CANbus 1

CAN messages on bus 1 are based on Appendix A from the ArtiTrax I report [1].

<table>
<thead>
<tr>
<th>ID</th>
<th>DLC</th>
<th>DATA</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x480</td>
<td>1</td>
<td></td>
<td>Joystick direction</td>
</tr>
<tr>
<td>0x481</td>
<td>1</td>
<td></td>
<td>Joystick speed</td>
</tr>
<tr>
<td>0x490:1</td>
<td>1</td>
<td></td>
<td>Amp1 requested front motor voltages</td>
</tr>
<tr>
<td>0x4B0:1</td>
<td>1</td>
<td></td>
<td>Amp2 requested rear motor voltages</td>
</tr>
<tr>
<td>0x4D0:1</td>
<td>1</td>
<td></td>
<td>Amp3 requested joint motor voltages</td>
</tr>
<tr>
<td>0x485</td>
<td>1</td>
<td></td>
<td>Amp1 start-up configuration parameters from joystick</td>
</tr>
<tr>
<td>0x4A5</td>
<td></td>
<td></td>
<td>Amp2 start-up configuration parameters</td>
</tr>
<tr>
<td>0x4C5</td>
<td></td>
<td></td>
<td>Amp3 start-up configuration parameters</td>
</tr>
<tr>
<td>0x494</td>
<td>1</td>
<td></td>
<td>Amp1 status of received configuration parameters, (24 if ok)</td>
</tr>
<tr>
<td>0x4B4</td>
<td>1</td>
<td></td>
<td>Amp2 status of received configuration parameters, (24 if ok)</td>
</tr>
<tr>
<td>0x4D4</td>
<td>1</td>
<td></td>
<td>Amp3 status of received configuration parameters, (24 if ok)</td>
</tr>
<tr>
<td>0x482</td>
<td>1</td>
<td></td>
<td>Three step speed setting displayed on control panel</td>
</tr>
<tr>
<td>0x483</td>
<td>2</td>
<td></td>
<td>Buttons on control panel</td>
</tr>
<tr>
<td>0x484</td>
<td>2</td>
<td></td>
<td>Lights</td>
</tr>
<tr>
<td>0x486</td>
<td></td>
<td></td>
<td>Warnings</td>
</tr>
<tr>
<td>0x489</td>
<td>1</td>
<td></td>
<td>Battery scale displayed on control panel</td>
</tr>
<tr>
<td>0x495</td>
<td></td>
<td></td>
<td>DMS redundancy</td>
</tr>
<tr>
<td>0x4E0</td>
<td>2</td>
<td></td>
<td>Amp1 motor current, approx 2A/unit</td>
</tr>
<tr>
<td>0x4E1</td>
<td>2</td>
<td></td>
<td>Amp2 motor current, approx 2A/unit</td>
</tr>
<tr>
<td>0x4E2</td>
<td>2</td>
<td></td>
<td>Amp3 motor current, approx 2A/unit</td>
</tr>
<tr>
<td>ID</td>
<td>DLC</td>
<td>DATA</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>0x4E8</td>
<td>2</td>
<td>0x0100</td>
<td>Amp1 motor voltage</td>
</tr>
<tr>
<td>0x4E9</td>
<td>2</td>
<td>0x0100</td>
<td>Amp2 motor voltage</td>
</tr>
<tr>
<td>0x4EA</td>
<td>2</td>
<td>0x0100</td>
<td>Amp3 motor voltage</td>
</tr>
<tr>
<td>0x700</td>
<td></td>
<td></td>
<td>Angle decoder, angle request</td>
</tr>
<tr>
<td>0x701</td>
<td>2</td>
<td>0x0100</td>
<td>Angle decoder, angle measurement</td>
</tr>
<tr>
<td>0x702</td>
<td>2</td>
<td></td>
<td>Left ultra sonic</td>
</tr>
<tr>
<td>0x703</td>
<td>2</td>
<td></td>
<td>Right ultra sonic</td>
</tr>
<tr>
<td>0x777</td>
<td></td>
<td></td>
<td>Debug used by Permobil</td>
</tr>
</tbody>
</table>

### A.2 CANbus 2

<table>
<thead>
<tr>
<th>ID</th>
<th>DLC</th>
<th>DATA</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x000</td>
<td>2</td>
<td>0x0100</td>
<td>CANopen startup</td>
</tr>
<tr>
<td>0x080</td>
<td>0</td>
<td></td>
<td>CANopen sync</td>
</tr>
<tr>
<td>0x601</td>
<td>8</td>
<td>0x2200180200000000</td>
<td>Encoder rear left, set to update position on sync</td>
</tr>
<tr>
<td>0x602</td>
<td>8</td>
<td>0x4002200000000000</td>
<td>Encoder rear left, speed request</td>
</tr>
<tr>
<td>0x602</td>
<td>8</td>
<td>0x4002200000000000</td>
<td>Encoder rear right, speed request</td>
</tr>
<tr>
<td>0x603</td>
<td>8</td>
<td>0x2200180200000000</td>
<td>Encoder front left, set to update position on sync</td>
</tr>
<tr>
<td>0x603</td>
<td>8</td>
<td>0x4002200000000000</td>
<td>Encoder front left, speed request</td>
</tr>
<tr>
<td>0x604</td>
<td>8</td>
<td>0x2200180200000000</td>
<td>Encoder front right, set to update position on sync</td>
</tr>
<tr>
<td>0x604</td>
<td>8</td>
<td>0x4002200000000000</td>
<td>Encoder front right, speed request</td>
</tr>
<tr>
<td>0x605</td>
<td>8</td>
<td>0x2200180200000000</td>
<td>Encoder joint, set to update position on sync</td>
</tr>
<tr>
<td>0x605</td>
<td>8</td>
<td>0x4002200000000000</td>
<td>Encoder joint, speed request</td>
</tr>
<tr>
<td>0x581</td>
<td>8</td>
<td></td>
<td>Encoder rear left, speed measurement</td>
</tr>
<tr>
<td>0x582</td>
<td>8</td>
<td></td>
<td>Encoder rear right, speed measurement</td>
</tr>
<tr>
<td>0x583</td>
<td>8</td>
<td></td>
<td>Encoder front left, speed measurement</td>
</tr>
<tr>
<td>0x584</td>
<td>8</td>
<td></td>
<td>Encoder front right, speed measurement</td>
</tr>
<tr>
<td>0x585</td>
<td>8</td>
<td></td>
<td>Encoder joint, speed measurement</td>
</tr>
<tr>
<td>0x181</td>
<td>8</td>
<td></td>
<td>Encoder rear left, position measurement</td>
</tr>
<tr>
<td>0x182</td>
<td>8</td>
<td></td>
<td>Encoder rear right, position measurement</td>
</tr>
<tr>
<td>0x183</td>
<td>8</td>
<td></td>
<td>Encoder front left, position measurement</td>
</tr>
<tr>
<td>0x184</td>
<td>8</td>
<td></td>
<td>Encoder front right, position measurement</td>
</tr>
<tr>
<td>0x185</td>
<td>8</td>
<td></td>
<td>Encoder joint, position measurement</td>
</tr>
</tbody>
</table>
REFERENCES


