Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Institut für Dynamische Systeme und Regelungstechnik

Hans Muster

Title

Subtitle

Bachelor's Thesis

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

First Supervisor
Prof. Dr. Second Supervisor

December 2016

IDSC-XX-YY-ZZ

Abstract

Abstract goes here

Keywords: First keyword, Second keyword.

Contents

Chapter 1

Hardware

1.1 Components

In the following section, we will shortly describe the relevant parts already built into the kart, as well as all other components we installed. We will also explain their importance and why we decided for these compontents.

1.1.1 Built into Go-Kart

The go-kart used for this project was a SinusiON, an electric kart manufactured by Rimo Germany. An electric kart offers an easier implementation of a throttle-by-wire system over more common petrol-fueled kart, as the basis for such a system is already in place. It's weight prior to modifying it was roughly 170 kg, making it much heavier than a petrol-fueled kart. The dimensions (l/w/h) are 2020mm/1390mm/600mm.

ACD 4805 Motor controller

Each of the electric motors is controlled by an ACD 4805 motor controller, mounted on top of the motor. The ACD communicates via CANOpen and allows for easy modification of parameters and performance curves. The motor controller was likely the most important in-built part, as the whole throttle-by-wire system was based on it.

Brake

Rimo offers a dual-circuit hydraulic disc brake system. A simple lever arm connects the brake cylinder to the brake pedal. This configuration requires a precise actuator, as the way difference between a mild and hard brake was minimal.

Steering

The go-kart's steering mechanism was realized with a bell-crank linkage. This setup resulted in a non-linear steering behaviour, which needed to be accounted for when configuring the power steering. The steering shaft was short and it's surroundings offered limited space, therefore the required steering servo needed to be small as well.

Battery

The main battery consists of 16 x 3.2 V LiFeMnPO4 cells and offers 40 Ah of battery charge. The nominal on-board voltage is 48 V. Because of the capacity of the battery, it makes a separate power supply for most of the additional electric actuators redundant. Only the power steering required 12 V and 90 A peak current. Because most converters do not support such a high peak current and the idea of capacitor seemed impractical, a small 12 V battery with a capacity of 3 Ah had to be installed in the front of the car.

Motor

The kart features two "PMS 100 R" 2.8 kW double-sided synchronous motors, each controlled by one of the two ACD 4805 motor controllers. The motors also offered regenerative braking, offering longer driving time and assisting in braking the car.

1.1.2 LinMot

In order to actuate the brakes a linear motor was used. The characteristics of being fast and precised make the motor suitable for various brake maneuvers. The motor is an electromagnetic drive in tube-form. The linear motion simulating a mechanical brake can be generated electrical without any form of mechanical interconnection between motor and motion. The linear motor is composed by two parts: the stator and the runner. The runner consists of a series of neodym-magnets, placed in steel tube. In the stator, the winding as well as the bearing for the runner, position detection and supervision of the motor fit in. The internal position sensors can dynamically transmit a position signal. Therefore position can be controlled real time. This guarantees a high level of safety and flexibility. In order to communicate with the motor, an LinMot motor-drive is employed. Target position, maximum velocity and acceleration can steadily be adjusted.

1.1.3 Power steering

Thyssenkrupp Presta offered a compact and powerful solution. The 3 Nm of rated torque was more than sufficient for steering the kart. The unit communicates via CANOpen, with TKP offering a Simulink Blockset for easy implementation.

1.1.4 DC-DC converter

In order to provide 24 VDC for powering the Microautobox and the linear motor drive, an SD-50C-24 DC-DC converter was used.

1.1.5 Microautobox (MABX)

The power steering manufacturer implemented a blackbox model of their unit in Matlab's Simulink. Their model was programmed to only run on dSpace's Microautobox. The Microautobox is a real-time system, used for performing fast function prototyping.

If needed, the MABX can easily be connected to a more powerful computer via the in-built ethernet connector. The MABX communicates with the go-kart's components via CAN. In it's basic configuration, the system does not support CANOpen, therefore an additional Simulink blockset had to be bought. The scarce time during the project justified the purchase of MABX, otherwise a much cheaper option could have been acquired. A less expensive option obviously calls for more programming and offers less plug-and-play, which surely would have slowed down our progress significantly.

1.1.6 cases/cables/adapters

A wide variety of cables, adapters were used in order to ensure seamless integration of the components into our system. Most of the cables we soldered ourselves were used for power supply. For the CAN communication we used a preexisting solution, where no soldering was needed. Previously, a RJ-45 cable connected the linear motor drive to the can network. A break out adapter was bought, in order to gain easy access to the CAN high and CAN low connectors.

A metal sheet case was bought and mounted in the back of the kart, where the DC-DC converter, the MABX and the Linmot motor drive were safely stored.

Chapter 2

Software

For this project, a variety of software was being used, namely dSpace Control Desk, LinMot Talk and Matlab Simulink. Without going into too much detail, we will elaborate on their use in this project and how they worked together.

Matlab is a numerical computing environment, with Simulink being a block diagram environment for multi domain simulation and model-based design.

dSpace Controldesk is a experiment software, used to develop and test operating ECUs. If offers data capture across different platforms and access to common Bussystems, including CAN and CANOpen.

LinMot Talk is LinMot's own drive-configuration software, allowing for easy access to the drive and controlling of the linear motor.

When the system is online, the Microautobox will receive commands from a higher layer, such as a simple rc controller or an A.I. autonomously controlling the vehicle. For this, a model of the communication interface needed to be set up in Simulink.

dSpace provided us with numerous Simulink blocksets, containing blocks of all necessary components, such as ADC, DAC and CAN. With these, a model can easily be formed and compiled. The compiler creates a C/C++ code, which then will be flashed onto the Microautobox. Controldesk can then be used to gain information about the state of the system's components and change values in real-time.

LinMot Talk was mainly used to find out the most efficient way to communicate with the linear motor and become familiar with it's characteristic. The software includes a configurable oscilloscope, where the variables and parameters of interest can be plotted. This allowed us to determine the optimal performance of the motor in the given circumstances. The software is also needed to flash the configurable firmware onto the linear motor's drive. This needed to be done once, as all of the parameters and commands can be changed afterwards via CANOpen.

Considering Matlab Simulink required the most work, we will now elaborate on our process of creating our Simulink model.

Chapter 3

Implementation

3.1 Steering

3.2 Braking

3.3 Throttle

Chapter 4

Communication

To handle the communication between all devices, a fast and easy to implement data exchange method was required. RIMO, as well as the other manufacturers of our components made use of the standardized industrial application CANOpen. Because CANOpen is based on CAN, we will first describe the CAN bus protocol.

4.1 CAN Basics

CAN stands for Control Area Network and consists of two main layers, namely the physical layer and the data link layer. CAN was developed in 1986 and is used for data exchange between different stations in a network, based on serial communication. Messages are received and transmitted via broadcasting, making every message available for all of the connected stations.

The rise in electronification in many parts of the industry called for simpler and more efficient means of communication. Extensive wiring still resulted in rather limited data exchange. A way out of this was presented by serial bit data exchange and connecting up all electronic control units to a single bus. Depending on the bus length, CAN offers up to 1 Mbit/s of data rate, while remaining robust and reliable, even in a noisy environment.

The physical layer consists of a twisted pair of wires, which can be shielded if required. The value of a bit was determined by the difference in voltage of the two wires. If the voltages are the same, the bit is recessive. If the difference is higher than 0.9 V, the bit is dominant. As both wires are affected by the same electromagnetic disturbances, their difference in voltage will not vary. The data link is therefore immune to electromagnetic disturbances. By twisting the wires, the magnetic field generated will also be reduced significantly.

To reduce reflections in the data cables with rates higher than 125kbit/s, it is recommended to terminate the ends of the communication lines with termination resistors with 120 Ohm.

Each CAN message contains the following structure.

4.2 CANOpen

4.2.1 Implementation

Chapter 5

Appendix

The following code is the definition of the bibliography entry of the document class IDSCreport [1].

```
@manual{IDSCreportClass,
author = {Andreas Ritter and Philipp Elbert and Christopher Onder},
title = {How to Use the {IDSCreport} {\LaTeX{}} Class},
language = {english},
howpublished = {Version 1.4.0},
organization = {Institute for Dynamic Systems and Control ({IDSC})},
address = {ETH Z\"{u}rich, Switzerland},
month = dec,
year = 2016
}
```

Bibliography

[1] A. Ritter, P. Elbert, and C. Onder, *How to Use the IDSCreport LaTeX Class*, Version 1.4.0, Institute for Dynamic Systems and Control (IDSC), ETH Zürich, Switzerland, Dec. 2016.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control

Prof. Dr. R. D'Andrea, Prof. Dr. E. Frazzoli, Prof. Dr. C. Onder, Prof. Dr. M. Zeilinger

Title of work:

Title

Subtitle

Thesis type and date:

Bachelor's Thesis, December 2016

Supervision:

First Supervisor
Prof. Dr. Second Supervisor

Student:

Name:	Hans Muster
E-mail:	muster@student.ethz.ch
Legi-Nr.:	??-???-???
Semester:	?

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf`

Zurich, 14.6.2017: ____________________