Technical Report

March 26, 2014

Anders Strand
Emil Taylor Bye
Petter S. Storvik
Odd M. Trondrud
Ole Bauck

Eksperter i Team
Norges Teknisk-Naturvitenskapelige Universitet

1 Abstract

The abstract is a tl;dr-type thing that *very* briefly summarizes the introduction, method, result and conclusion.

1 Contents

2 Introduction

Introduction goes here...

3 Concept Description

Concept description goes here

4 SWOT

SWOT analysis of the project goes here. I guess one of our weaknesses is that everything is run on top of https but then again that might be one of our strengths depending on the area for which one wants to apply the system. Also what about monetization? I mean we've got like, no prospects for monetization. Is that bad? Should we write about that?

From Wikipedia:

> Setting the objective should be done after the SWOT analysis has been performed. This would allow achievable goals or objectives to be set for the organization.

Well we did not do that.

Strengths

- everything is built on already existing technologies
- mutability/modularity/expandability or whatever
- robots are pretty cool and I've heard the internet is hip these days

Weaknesses

- http is kinda slow I guess? for video at least. Something about how this limits the possible real-time usage of the system
-

Opportunities

- nothing like this exists and being first is always cool
-

Threats

- I think I speak for the entirety of the group when I say that we'd rather be on a beach somewhere sippin' piña coladas or wherever, like in an igloo or something, than doing *work*.

5 Method

5.1 Vehcle

To demonstrate the platform it was decided to make a vehicle controlled remotely over internet. This vehicle had to be able to drive, turn, use a rabotic arm and send camera feed to the controller. Due to the possibility to expand this later with more complex functionality, the program was written in C++. The following was used to make this vehicle:

- Raspberry Pi
- Dynamixel servomotors
- USB2Dynamixel
- Dynamixel SDK

Rasppberry Pi (Pi) is a small single board computer. It has two USB inputs, ethernet, HDMI and gpio pins. This makes it the perfect prototyping computer for this kind of projects. The Pi is running its operating system from a SD card, and the OS is RaspBian which is a debian based linux distro. To controll the Pi a regular laptis is used. This is achieved by using a SSH server on the Pi and a SSH client on the computer. SSH (Secure SHell) is a protocol that allows one computer to remotely controll another via command line.

Dynamixel servomotors[1] are motors that allows for precise controll of angle and velocity. These motors are controlled by three wires, power, gnd and signal. WRITE SOME MORE HERE!

USB2Dynamixel is a USB device that allows easy controll of dynamixel servo motors. This USB requires no driver installation when running linux, and since RaspBian is a linux distro this simplifies things. The USB2Dynamixel is inserted into one of the USB ports on the Pi and the servor motors are connected to the device.

Dynamixel SDK[2] is a programming library for controlling dynamixel servo motors. This library is available for Windows, Mac and Linux and easy to run on the Pi. Since the library is written in C it is easy to use it in this C++ project.

[1] `http://www.robotis.com/xe/dynamixel_en`
[2] `http://support.robotis.com/en/software/dynamixelsdk.htm`

5.1.1 Wiggle It

The first step is to make a servo motor move. Since the USB device is linux compatible and the library also works in linux the Pi was used from the begining of the project.

Figure 5.1: fig:1servo

To make the servo motor move an example c-file form the library was used[3]. For this to work there are som important parameters which should be known:

- ID - Each motor has its own ID. This way its possible to control which motor that should get the command.
- Port - Which port the USB2Dynamixel is connected to. This is used to set up the communication to the motors.
- Mode - The servo motors can run in two different modes. Endless turn (like a wheel) and start2stop (from start position to specified position).

The servo motors used for this test had IDs 7 and 12, and can be found on the side of the servo. In linux the USB device can be found under /dev/usb, and using the Pi this device was /dev/usb0. By replacing this parameters in the example file two servos was wiggled.

5.1.2 Driving

The next step is to make four wheels cooperate!

5.1.3 Future problems/challenges

What happens if some of the wheels is disconnected? The answer for this question will not be covered in this report because of the timeframe, but it is an important issue that has to be resolved. The optimal solution is to find an algorithm that calculates the speed that the remaining wheels, so that the vehicle can obtain normal function even in failsafe mode. This will most likely depend on the surface under the vehicle.

We should have like, a graph or drawing of the finished system here shouldn't with the involved components and actors and what gets transferred between them. Like those drawings Anders have been keeping safe in his lockerbox or whatever.

[3]LOCATION TO EXAMPLE FILE