

Fall Semester 2015

Line following robot

Group 2

2. Semester IT-Technology

Group members: Benjamin Nielsen - Henrik Jensen - Martin Nonboe - Nikolaj Bilgrau

Supervisor: Jesper Kristensen - Steffen Vutborg

Title:

Line following robot

Project Period:

2. Semester | Spring semester 2016

Projectgroup:

Group 2

Medvirkende:

Benjamin Nielsen

Henrik Jensen

Martin Nonboe

Nikolaj Bilgrau

Supervisor:

Jesper Kristensen

Steffen Vutborg

Pages: TBD

Appendices: TBD

Completed TBD

Introduction

Project is written by:

Benjamin Nielsen

Henrik Jensen

Martin Nonboe

Nikolaj Bilgrau

Table of Contents

1	Requirements specification	1
2	Hardware section	2
2.1	Description of the hardware structure and functionality	2
3	Software section	3
3.1	Description of the software structure and functionality	3
4	Test	5
4.1	Test	5
5	Conclusion	6
6	Appendices	7
6.1	Group collaboration agreement	7
7	List of references	8
	List of Figures	9
	List of Tables	10

Glossary

A/C Air Condition

CPU Central Processing Unit

I/O Input/Output

PCB Printed Circuit Board

UPS Uninterruptible power supply

Requirements specification

1

The following section will describe the specific requirements that have been decided to fulfil to the general requirements as shown in the project description.

- Project must include light sensors
- Implement motor control
- Should make use of the Pic32 MCU
- Software will be written in MPLABX
- Autonomous operation
- The product must make use of feedback concept e.g. a PID algorithm
- A function & performance test is to be conducted

Hardware section 2

2.0.1 Hardware diagram

2.1 Description of the hardware structure and functionality

2.1.1 Selection of sensor

Name	QRE1113 board	OPB706A	OPB704
Max sensor distance	3mm	1.27mm	3.8mm
Forward current	50mA	20mA	40mA
Mounting	On print	THT	In casing
Price	19.43DKK	26.90DKK	42.55DKK
Notes			

Table 2.1: Table of a selection of sensors

TBD Beskriv sensorer og hvorfor vi har valgt denne

2.1.2 OPB704 Sensor

Sensor choice for the line following robot will be the OPB704. The choice was logical because of the availability. It was confirmed in the brainstorming process that the sensor was reliable and had a wide range of applications. 3D printing the mount was ideal for the purpose of the sensor and the task ahead.

TBD Beskriv hardware struktur og funktion samt alle underdele

3.0.1 Software diagram

TBD Softwarediagram

3.1 Description of the software structure and functionality

TBD Softwarebeskrivelse og underafsnit

3.1.1 Description of the PID controller

A PID controller continuously calculates an error value as the difference to a reference point and a measured process variable.

PID is an abbreviation for a proportional-integral-derivative controller, it is a control loop feedback mechanism. The controllers job is to minimize the error value for the given devices running time. In the case of this project the reference point is the line and the PID will power up the engines to steer accordingly to said reference point.

$$F(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Proportional control(P)

The proportional term produces an output value that is proportionally related to the current error value, this value can be adjusted by modifying the error by a constant (K_p). A high proportional gain results in a large change in the output for a given change in the error.

$$P_{\text{out}} = K_p e(t)$$

If the proportional gain is too high, the system can become unstable. Contrarily, a small gain will result in the device adjusting too slowly, which decreases overall efficiency and in the case of this project, it will end up being detrimental to the steering accuracy.

Integral control(I)

The integral controller is contributing proportionally to both the magnitude of the error and the duration of the error.

The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously.

The controller output equals the accumulated error multiplied by the integral gain(K_i)

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau$$

The integral term accelerates the movement of the process towards the reference point. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the reference value.

Derivative control(D)

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d . The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d .

The derivative term is given by:

$$D_{\text{out}} = K_d \frac{de(t)}{dt}$$

The derivative action predicts system behaviour and utilizes this to improve the settling time and stability of the system. An ideal derivative is not causal, so that implementations of PID controllers include an additional low pass filtering for the derivative term, to limit the high frequency gain and noise.

Test 4

4.1 Test

Conclusion 5

TBD Konklusion

Appendices 6

6.1 Group collaboration agreement

6.1.1 Contact Information

TBD contact info

6.1.2 Workflow

- Every friday after 12:00 is expected work consisting of three hours.
- If you aren't able of attending for scheduled study day. - Notice must be given to the project team.

6.1.3 Milestones and goals

TBD Milestones

6.1.4 Deadline

- Hand in June 7th.

List of references 7

TBD list of references

List of Figures

Page

List of Tables

2.1	Table of a selection of sensors	2
		Page