Technical University of Denmark

André D. B. Christensen (110033)
Jacob A. F. Pedersen (120374)

Eyebot - A Line Following Robot Based On Camera Tracking

4th Semester Project in Control Theory

Eyebot - A Line Following Robot Based On Camera Tracking

This report was prepared by:
André D. B. Christensen (110033)
Jacob A. F. Pedersen (120374)

Advisors:
Henrik Møller, DTU Ballerup Campus

DTU Diplom
Lautrupvang 15
2750 Ballerup
Denmark
Tel: +45 3588 5088

diplom@diplom.dtu.dk

Project period: September 2013 - December 2013

ECTS: 10

Education: BEng

Field: Electrical Engineering

Copyrights: ©André D. B. Christensen & Jacob A. F. Pedersen, 2013

Contents

Chapter 1

Introduction

Autonomous robots are robotic devices capable of performing tasks based on inputs from various types of sensors, without human interaction. Much research in the recent years has focused on the development of autonomous cars (also known as Unmanned Automobiles) which can drive without interaction from the driver, but just by sensing the environment. This is an interesting field, as it may lower the number of traffic accidents and especially reduce the traffic congestion in bigger cities by providing a better traffic flow.

In this present project we designed and constructed an autonomous robot, capable of following a line by using a mixture of cameras and IR sensors for gathering input about the surroundings. Close attention was paid to the use of camera sensing and image processing for detecting and extracting the line. The traditional method has been to use an array of photo-sensors, but due to the low resolution, an alternative method is necessary for archiving a better stability.

This paper outlines the work done, and shows how a camera can be used instead of a traditional array of photo sensors. Further more, as an additional feature, it is explained how the live image data is streamed to a computer, for a live preview of what the 'robot sees'.

Our robot may in the following chapters be referred to as 'the robot', 'the system' or simply 'Eyebot'. TODO More attention to traffic accidents, traffic flow and congestion.

1.1 Subsystems and Block Diagram

The robot consist of several subsystems, which have different responsibilities. The block diagram in figure **??**, outlines the basic components of the system with connections betweens them.

As seen on the block diagram, the main processor of the system is an ARM-based Raspberry Pi computer. The Raspberry Pi first of all acts as the overall state machine when driving the track. It knows when to follow the line, go to the wall, rotate and so on (more on that in chapter 6). Moreover the Raspberry Pi takes care of capturing images from the camera, processing the images, and calculating the error signals when following the line. As an additional feature, the live image data when following the line, are transmitted over WiFi to a laptop, for live preview of the robot's processed and annotated image. The image capture and processing is discussed in detail in chapter 2.

Below the Raspberry Pi, an Atmel XMEGA192C micro-controller is responsible for powering and controlling the motors. The XMEGA is instructed directly by the Raspberry Pi, by commands received over the I2C (Two Wire Interface) bus. Equally important the XMEGA reads analog signals from the three distance sensors, and makes them available for the Raspberry Pi through the I2C bus. The motor-controller is discussed in chapter 3 and the distance sensors are discussed in chapter 4.

As an additional feature, the robot has an array of LEDs mounted on its bumper, and a buzzer for indicating state changes. Those are powered by an MCP23016 I/O expander chip, that communicates with the Raspberry Pi using I2C. Tis subject will not receive any more attention in the report than this.

1.2 Requirements Specification

Autonomous robots are robotic devices capable of performing tasks based on inputs from various types of sensors, without human interaction. Much research in the recent years has focused on the development of autonomous cars (also known as Unmanned Automobiles) which can drive without interaction from the driver, but just by sensing the environment. This is an interresting field, as it may lower the number of traffic accidents and especially reduce the traffic congestion in bigger cities by providing a better traffic flow.

In this present project we designed and constructed an autonomous robot, capable of following a line by using a mixture of cameras and IR sensors for gathering input about the surroundings. Close attention was paid to the use of camera sensing and image processing for detecting and extracting the line. The traditional method has been to use an array of photo-sensors, but due to the low resolution, an alternative method is necessary for archiving a better stability.

This paper outlines the work done, and shows how a camera can be used instead of a traditional array of photo sensors. Further more, as an addtional feature, it is explained how the live image data is streamed to a computer, for a live preview of what the ‘robot sees‘.

1.3 Requirements Specification

1.3.1 Assumptions

We assume an average speed for the robot of 1 m/s. By looking at the line, we found that it should be reasonable to make corrections every 0.1 m. This gives an maximum cycle time of 100 ms. Cycle time includes capturing and processing of the image, calculation of the error and correction value, transmitting the corrective actions to the motors, and waiting for the motors to respond and react. By tilting the camera forward, we might be able to detect future obstacles, and take required actions such as slowing down.

1.3.2 Formal requirements

1. The robot should be able to follow the track on first floor. This includes
   - Make a decision on which way to turn when seeing the line first time
   - When a short peace of tape intersects the line, the robot should turn 90 degrees to the left and drive towards the wall. It should stop 20 centimeters from the wall, turn 180 degrees (TBC) and go back and follow the line as usual.
   - When a short line is sensed on the left side of the line, the robot should speed up, and try to go as fast as possible for the rest of the track.
   - At the end of the track, the robot should drive to the wall, keeping a distance of 20 centimeters to the wall, and then follow the wall to the end of the track.
2. The robot should be able to drive at a maximum speed of 1 m/s (TBC).
3. The robot should make corrections every 0.1 m (TBC), which is every 100 ms (TBC) at 1 m/s.
4. The line should be sensed using a camera and image processing (TBC).
5. A Raspberry PI should be the main controller in the system.
6. The Raspberry PI should capture images of the floor, analyse the image, and calculate the error.

7. The motors should be controlled by an AVR microcontroller, that interfaces the Raspberry PI using I2C (TWI).

8. The speed of the motors should be controlled using PWM signals through a H-bridge.

9. The distance to the wall should be measured using a distance sensor in the front, and one or more on the left side of the robot.

1.4 Time Schedule

CHAPTER 2

Tracking And Processing The Line Using Image Processing

In this section the methods for capturing images and performing the nessesary processing are explained in detail, with emphasis on how the image is processed to produce an error signal for the control system.

2.1 Capturing The Image

As explained in the introduction, the camera used for this project is a Sony PS3 Eye camera, which is based on OmniVision's OV543 camera sensor. All image capture and processing is carried out in the Raspberry Pi running the Raspbian Linux distribution.

The PS3 Eye camera is supported directly by the Linux kernel, and can be controlled by user software through the Video4Linux driver interface. The PS3 Eye camera was choosen for this project, because of its specification and low price. The camera is capable of shooting 125 frame per second, why it is perfect for computer vision applications (what is actully also was made for).

When the robot starts, the camera is set up to deliver 30 frames per second, in gray-scale colors. As part of the initialization, a callback is given to the camera module, and is called every time a new frame is ready for processing. All source code for initialization and setup of the camera, is implemented in 'camera.h' and 'camera.c' (see appendix).

..

..

..

FPS, size, camera angle, camera height. Playstation3 Eye Camera, OV534 sensor, Video4Linux driver, memory mapped files for faster access. Limitations of the frame rate.

2.2 Identifying and Extracting the Line

Whenever an image has been captured and is available in the memory, it is time to do the image processing and extract the line from the background. This is done using the steps listen below.

- Contrast and brightness is adjusted
- The image is 'sliced' into six horizontal slices
- For each slice, a histogram is calculated, and from that a optimum thresholding value
- Finally each pixel in each slice is marked as LINE if below the threshold value, or marked as FLOOR if above the threshold value

2.2.1 Contrast and brightness

The contrast and brightness adjustments are simple point-processing operations, which can be describes with the following equation.

$$g(x, y) = a \cdot f(x, y) + b \tag{2.1}$$

2.2.2 Histogram

Figure: Image before thresholding, histogram of the same image (use MATLAB's imhist and imread)

2.2.3 Slicing

Figure: Image with slices marked

2.2.4 Optimum thresholding

Moving average filter (16-length) Figure: Thresholded image, histogram illustration

2.3 Calculating the Error Signal

Center of middle mass, multiple points, mass

Figure: Image with error points

$$x_c = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2.2}$$

$$y_c = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{2.3}$$

2.4 Performance Considerations and Possible Improvements

Frame rate, further processing, light

CHAPTER 3

Controlling The Motors

3.1 The DC-motor

3.1.1 DC-motor

The DC-motor is a essential part of the robot. The motors used in this project is the Faulhaber 2233012s which is a 12 Volt armature motor. To succeed the tasks given in the assignment is it very importent to gain control over the motor by understand the datasheet that describes the motor. The motors dose not only supply the robot with driving force but is also the steering part of the robot.

3.1.2 Gears

To give the robot more torque a gear system has been implemented in the motors. The cost of high torque through a gearsystem is a reduction in rotational speed. The gearing factor that is used is $\frac{1}{17.2}$ which means that the output shaft only turns one cycle every times the motor itself have turned 17.2 cycles. The gear system do have some disadvantages by giving the motor nonlinearity like backlash and deadlocks.

3.1.3 Transferfunction for DC-motor

In assignment 1 from the control theory course we calculated a transfer function for at DC-motor. By using this knowledge and theory we a able to calculate a transfer function for the DC-motor used in this project. The transfer function for the DC-motor is needed for making simulations of the hole system later on.
All motor data is fund in the datasheet and the weight of the robot is fund by

weighting the robot on a digital precision scales and is fund to be 1400 g or 1.4 Kg.

The general transfer function:

$$G(s) = \frac{\theta_m(s)}{E_a(s)} \Rightarrow \frac{\frac{k_t}{R_a J_a}}{s + \frac{1}{J_a}(D_a + \frac{K_b K_t}{R_a})} \tag{3.1}$$

The transfer function for the robot:

$$G(s) = \frac{\omega(s)}{E_a(s)} = \frac{\frac{1633}{23.43}}{s\frac{1}{23.43} + 1} = \frac{69,69}{0,04268 + 1} \tag{3.2}$$

Where 0.04268 is the time constant.
The step respons on the DC-motor looks like this:

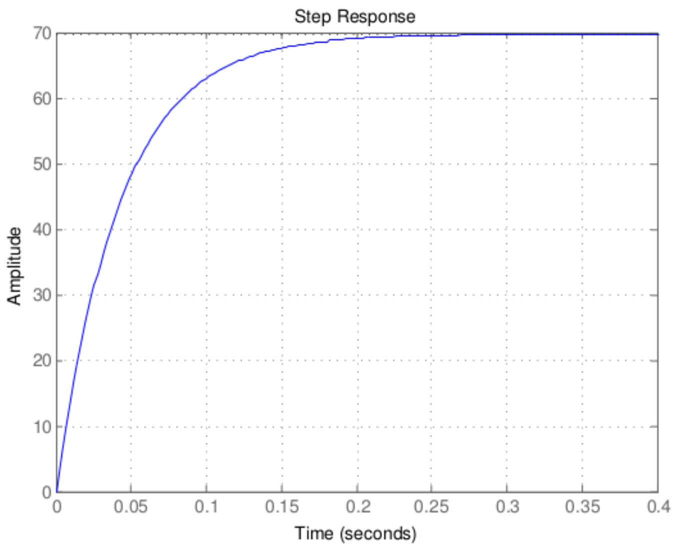

Figure 3.1: Step respons for DC-motor

3.2 Microcontroller

The MCU (Microcontroller) is the first part of the link between the sensor input and the DC-motor. In the MCU a translation is made so that the DC-motor can be controlled using simple I2C (Two way communication) commands from the sensor device. A controller is also implemented in the MCU so that the robot is able to run straight without any further instructions from the sensor device.

The MCU decodes the command form the sensor devise and then extracts informations about direction and speed. These informations is then sends down to the H-bridge and ends as Voltage in the DC-motor.

3.2.1 The MCU

The MUC used in the robot is an Atmel ATxmega192c3 a 8bit low-power microcontroller with 192kbyte flash memory running a clock frequency at 32 MHz. The choice of this MCU is based on the its speed performance and its high level of functionality features. The first MCU used in the robot was a Atmel Atmega 32 but after a brief discussion in the group and with the project supervisor we decided to change the MUC because we previously worked with the Atmega32 and wanted to explore more recent technologies and the opportunity to work with the SMD (surface mounted device) method.

3.2.2 Programming

The MUC is programed using Atmels studio 6.1 and the program language used is C. The ASF(Atmel Software Framework) libraries has be used in great extent. For more informations here about visit:
www.atmel.com/tools/avrsoftwareframework.aspx
The source code for the MCU can be found in the appendix.

3.3 H-bridge

The H-bridge receive three line per motor from the MCU The first line controls the speed with a PWM signal and the to other lines control the rotational direction of the motor.

3.3.1 PWM Pulse-width modulation

PWM (Pules-width modulation) is a method frequently used in electronics. The signals intensity it controlled by changing the duty cycle for a square wave signal with a predefined frequency. The controllable range is from 0% to 100% The frequency of the square has been chosen to be 20kHz so that is it over the human hearing range.
The figures below shows how a 25% and a 50% PWM looks between the MCU and the H-bridge. The duty cycle output of the MCU is about 15 % and 82% The reason for that is that the controller in the MCH manipulates the PWM, more about this in the Feedback and Control Loop section.

resources/Scop/PWM_motot_right_PWMin25.png

Figure 3.2: PWM with input at 25%

resources/Scop/PWM_motot_right_PWMin50.png

Figure 3.3: PWM with input at 50%

3.3.2 Direction control

To control the direction of the motor the H-bridge receives two lines A1 and A2 these two lines dictates the rotational direction of the DC-motor.

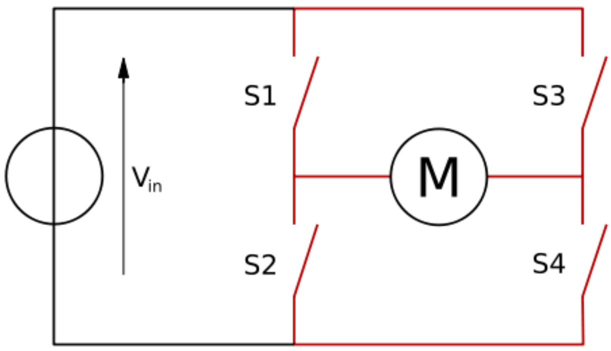

Figure 3.4: H-bridge

When line A1 is high and line A2 is low the corresponding switches S1 and S4 is closed so that the current is running in one direction. when A1 is low and A2 is high the current is running the other way and the motor it turning the other way around.

below is a truth table that describes the functionality of the H-bridge. The EN pin is the PWM input.

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 3.5: H-bridge truth table

3.4 Encoders

The DC-motor includes two encoder outputs which is used to determine position, velocity and direction of the DC-motor. Each encoder outputs 15 pulses per round and after the gear is the resolution $15 \cdot 17.2 = 258$ pulses per round. The two encoder signals is phase shifted at about 90° and that gives us the opportunity to determine the rotational direction of the motor by looking at one encoder signal and then check if the other one it high or low. This method is illustrated below in figure 3.6.

resources/encoder_atmel.png

Figure 3.6: Encoder illustration from atmel Application Note AVR 1600 aboute Quadrature Decoder

On figure 3.7 we see the actual encoder output from the robot. The phase is about 100° between the two encoders. The velocity is found by sampling the

resources/Scop/encoder.png

Figure 3.7: Encoder output from Motor to MCU

encoder pulses over a time period of 10ms, by counting the pulses during this time and multiply it with a constant the velocity of the motor is given. Also the position between the two motors can be find by using two encoders on each motor. First we look at one motor encoder count and then match up the other motor so that the encoder count from the to motors fits, then we know the exact position of the robot. We use these method in the position controller.

3.5 Feedback and Control Loop

The control loop containing the motor,MCU and encoder can be described as a unity feedback loop.

resources/feedbackloop.png

Figure 3.8: Unity feedback loop

The MCU have three different stats with three different controllers so that it can adapt to the given situation during the track. The three states are:

- 1. Straight
- 2. Position
- 3. Speed

The selection of which current stat the MCU should be in is determiant by the raspberry pi(sensor system).

3.5.1 Straight state

This state is designed to make the robot go straight without any corrections from the sensor system.This it done by given both motors a reference speed and then aligning them after the outputs from the encoders by adjusting the speed to each motor.

3.5.2 Position state

3.5.3 Speed state

This state is used when the robot follows the line by adjusting after instruction from the sensor system. A P-controller it used and the encoders have no influence on the control of the motor.

CHAPTER 4

Measuring distance to wall

4.1 Practice use

How we inplemented it to solving obstacles around the track. Why do we use two of them?

4.2 IR sensor

4.3 Converting using an ADC

ADC in microcontroller. wdwdwdwd What we do with the data dist –¿ cm converting.

CHAPTER 5

Communication between devices

CHAPTER 6

Following The Track

6.1 Requirements for the track

Line, going straight forward, precise turns (position control), measuring distance to the wall in the front, measuring distance in the side

6.1.1 State machine

Go to line, follow line until crossing, turn towards the wall, go straight until the robot are 20 cm from the wall, turn 135 deg, go straight until the line is visible again, follow the line until a tape marking, follow the line with higher speed, stop af next tape crossing, go to the wall and stop 20 cm from it, turn 90 deg follow the wall...

CHAPTER 7

Live streaming of video

7.1 Server-client model

7.2 Viewer / client

CHAPTER 8

Simulation and Testing

8.1 SIMULINK

Appendices

APPENDIX **A**

First Appendix

A.1 Section in Appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

A.2 Another Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam.

Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.