KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle, 15 credits

Mechatronic Bedbug Attractor

Attracting bedbugs using the principles of human breathing

SIMON LILJA
MAJA ÅSTRAND

Stockholm, Sweden, 2022

Mechatronic Bedbug Attractor

Attracting bedbugs using the principles of human breathing

SIMON LILJA

MAJA ÅSTRAND

Degree Programme in Mechanical Engineering
Date: May 22, 2022

Supervisor: Varun Venkataraman
Examiner: Martin Edin Grimheden
School of Industrial Engineering and Management
Swedish title: Mekatronisk väggluslockare
Swedish subtitle: Användandet av principen för mänsklig andning för att locka vägglöss

Abstract

Bedbugs are insects which exclusively feed on the blood of humans and other large animals. Bedbug infestations are a growing problem in modern society and some of the current solutions involve humans acting as bait to lure the bedbugs out of their nests. Research indicates that bedbugs are attracted to, among other things, the increased levels of carbon dioxide (CO_2) that humans emit. This project aimed to create a machine that was able to simulate human breathing to a point where it could be used as a human substitute to a human during treatment for bedbug infestations. The prototype was based on diaphragmatic breathing and a pump mechanism was built which breathes air in and out. The machine's purpose was to manipulate the air by adding a specific amount of CO_2 and humidity. To control the levels of CO_2 and humidity, a control system was designed. The system was tested with the goal of being able to maintain a specified level of CO_2 and humidity over a longer period of time. Tests found that CO_2-levels were possible to regulate with relative ease, however the chosen components for the humidity-system was found to not be effective in regards of rapidly increasing the humidity to the levels that can be found in human exhaled air.

Keywords: Mechatronics, Simulated breathing, Bedbugs, Carbon dioxide, Solenoid valve.

Sammanfattning

Vägglöss är en insekt som exklusivt livnär sig på blod från djur och människor. Vägglössangrepp är ett växande problem i samhället och vissa nuvarande lösningar involverar att en människa agerar lockbete för att locka ut vägglössen från sina gömställen. Forskningen pekar på att vägglöss attraheras av bland annat de ökade halterna av koldioxid (CO_2) som utsöndras av människan. Detta projekts ändamål var att skapa en maskin som simulerar mänsklig utandning för att kunna användas som ett substitut för en människa under en behandling av ett vägglössangrepp. Prototypen har baserats på diafragmaandning och en pumpmekanism har byggts som andas in och ut luft. Maskinen ämnar manipulera den inandade luften med en önskad mängd CO_2 och fukt. För att kontrollera de önskade nivåerna designades ett kontrollsystem. Detta har sedan testats med målet att kunna ha en önskad och stabil nivå av CO_2 och fukt under en längre period. Tester fann att CO_2-nivåerna gick att reglera med ett relativt enkelt kontrollsystem men att det valda luftfuktningssystemet inte var effektivt nog för att höja fuktnivån till den som återfinns i mänsklig utandningsluft.

Nyckelord: Mekatronik, Simulerad andning, Vägglöss, Koldioxid, Magnetventil.

Acknowledgements

We would like to thank our supervisor, Varun Venkataraman, for his guidance through the project and for his valuable input. We would also like to thank the project assistants, Algot Lindestam and Tore Malmström for their assistance and general company in the laboratory.

Contents

Acronyms

ASC	Automatic self-calibration
CO_2	Carbon dioxide
dm^3	Cubic decimeters
°C	Degrees Celsius
FRC	Forced recalibration
K	Kelvin
ml	Milliliters
ms	Milliseconds
NC	Normally closed
NO	Normally open
ppm	Parts per million
RH	Relative humidity
V	Volts

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is part of a degree project in Mechatronics at KTH, Royal Institute of Technology. In this chapter a brief introduction for the project will be presented.

1.1 Background

Bedbugs are insects that feed on animal blood and the main factor for a bedbug's survival chances and reproductive opportunities are access to animal blood. Bedbugs can be found in different places but most likely close to a feeding opportunity [1].

Bedbug infestations are an increasing problem in society, in the recent 20 years the number of infestations have increased exponentially in many major cities. This is a result of, among other things, pesticide-resistance [2]. The effects of an infestation for humans involve both psychological distress and immune responses that can cause physical discomfort. Most of the infestations are found in big housing, communal households, and within the hospitality industry. The problem has a major economical impact on society, including both the obvious costs for pest control services, but also social reputation. [1].

Research has shown how bedbugs are attracted to the human aspects of breathing and, most significantly, the amount of CO_2 being released during the human breathing cycle [2]. One way of treating an infestation of bedbugs today is to use *diatomaceous earth*, a white soil which has the purpose of drying the bedbugs skin out causing the bedbug to perish after some time. Diatomaceous earth is placed closed to where the ongoing infestations is taking place in hope for the bedbugs to encounter the substance. However, for the bedbugs to encounter the substance an enticement outside their nest is required and in most cases that involves the presence of a human [3].

1.2 Purpose

The purpose of this project is to make a machine that attracts bedbugs. It is known that bedbugs are attracted to CO_2 and other aspects of the human respiratory system [2]. This project is inspired by those findings and will attempt to design a machine that is able to control the release of CO_2 similar to how the human respiratory system releases the CO_2. To further resemble human breathing the factor of adding humidity to the exhaled air has been taken into consideration. Additionally, with a machine that is able to control the CO_2 release, research on what levels of CO_2 bedbugs are being attracted to can be performed. It should also be possible to use the machine as a replacement for a human while treating an infestation. This report aims to answer the following scientific questions about said machine:

- How could a machine be designed to resemble human breathing?
- How can a control system be designed to ensure that the machine can reach and maintain a desired level of CO_2 release?
- How could a CO_2 release system be designed to resemble CO_2 release similar to the human respiratory system?

1.3 Scope

This project focuses on the practicalities of designing a machine to simulate some of the key aspects of the human respiratory system being attractive for bedbugs; namely the inhalation and exhalation of air with the increased levels of CO_2 together with added humidity [4]. Other features of the human respiratory system have not been taken into consideration. Most notably the machine will not be able to heat the air, which is not only an important factor on its own but also important for the airs ability to accept humidity in greater quantities.

Furthermore this project will only try to find a singular solution in order to control the mechanical movements and the CO_2 release aspects. No attempts to fine-tune any of the parameters beyond the CO_2 level in order to be more or less attractive to bed bugs will be made.

1.4 Method

Previous research about relevant attractants for bedbugs have been analyzed and have played a significant role in the machine's intended design, function and purpose. To resemble human breathing a pumping mechanism inspired by diaphragmatic breathing was built and information was gathered about relevant components of the human breath. To control these components of the "inhaled" air, various subsystems have to be constructed:

1. A CO_2 release system controlled by a solenoid valve.
2. A humidifier system.
3. A control system for controlling CO_2 release with feedback data from a sensor that measures CO_2.

These systems will then be tested in controlled experiments to verify their functionality and to attempt to answer the questions posed in Section 1.2. A further depth into how these systems will be implemented is given in Section 3.

Chapter 2

Theoretical Background

This chapter will present the theoretical background required for the projects aim.

2.1 Bedbugs

In order to demonstrate the purpose of the project, information about bedbugs, bedbugs infestations, and bedbug attractants have been gathered. There are a lot of research papers and information to be found regarding what bedbugs are attracted to. This report will primarily focus on the CO_2 that can be found in human breath.

2.1.1 General

Bedbugs belong to the Cimicidae family of insects. This family of insects is characterized by their dependence on animal or human blood as a food source. Bedbugs go through five life stages that span from their hatching to a fully grown insect, a process that takes around 37 days. Figure 2.1.1 shows an overview of a bedbug life cycle. In order to grow larger bedbugs need to shed skin and in order to do so they need to have regular blood meals. Bedbugs predominantly stay hidden in cracks and cervices, typically close to a feeding host. Bedbugs are most active during night, in search for their feeding host. They are most likely to reproduce shortly after a blood meal. To stay healthy and to be able to reproduce they need to feed every three to seven days. The female bedbug's ability to produce eggs is dependant on the access to blood meals and on an average they lay 1 to 7 eggs per day for around 10 days and can produce between 5 to 20 eggs from a single meal. They can live without feeding for up to 70 days before dying from starvation [5].

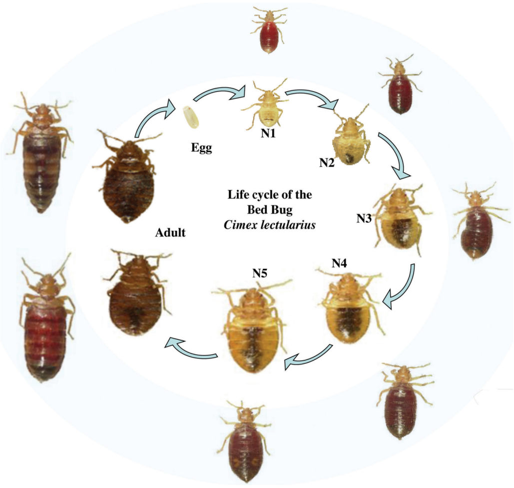


Figure 2.1.1: An overview of a bedbug's life cycle [5].

2.1.2 Infestations

An infestation can source from a single mated female bedbug without having a male present. When the female's offspring's have grown into reproductive active males they are able to mate and keep the infestation active [5]. The density of the infestations increases together with an increased density of humans or other potential feeding hosts such as pets. Some of the biggest factors for a large infestation are brief occupancy and high density accommodations. This could for example include areas with a high number of travelers or guest workers [1]. Different solutions to prevent an infestation of bedbugs to grow and preclude bedbugs being able to reach the host is available. For example, mattress protectors. Figure 2.1.2 shows a bedbug trap aimed at preventing a bedbug from climbing up a bed leg. Depending on the size of the ongoing infestation diatomaceous earth can be used as mentioned in 1.1. If the diatomaceous earth method does not work, heat treatment may be necessary. Heat treatment involves heating up the room or area of the infestation for sufficiently high temperature in order to kill the bedbugs. However this solution can damage furniture and walls [3].

Figure 2.1.2: A bed leg bedbug trap [3].

2.1.3 Attractants

CO_2 and various chemicals emitted from animals are all thought to be components that attract bedbugs. The emitted amount of CO_2 has been shown to be the key component for attracting bedbugs [6]. Bedbugs are able to find their host by signals of increased CO_2 in the room, increased levels of CO_2 has been shown to stimulate hungry bedbugs to begin searching for the host [5]. It has also been shown that attracting bedbugs using CO_2 is a much better method to determine the size of the infestation than using any visual cues. [6]. It has been shown that bedbugs have shown interest in traps where 1 ml/min of CO_2 have been released, in these cases the CO_2 source is from dry ice [2].

2.2 Human Breathing

To be able to resemble the principle of how human release CO_2 through breathing the mechanical aspects have been looked into including the general principle of diaphragmatic breathing.

2.2.1 Chemical Composition

The exhaled air of a human breath contains somewhere in between 4% and 5% CO_2 which is equivalent to 40 000 – 50 000 Parts per million (ppm). It also contains 0.9% water vapor and the exhaled air of a human is completely saturated with water, meaning the relative humidity is 100% [4]. A human exhales around 500 Milliliters (ml) of air in each breath [7]. In contrast the atmospheric air consists of, on average, around 400 ppm CO_2 while the ambient air in a room usually reaches up to (and above if ventilation is bad and the room is occupied) 1 000 ppm CO_2. Depending on the quality of the air in a room, the CO_2-level in an exhaled breath of air is between 50 – 125 times higher than the surrounding indoor air [8].

In table 2.2.1 a more digestible way of treating these numbers are shown by their respective absolute volumes given a total volume of half a liter, or 0.5 Cubic decimeters (dm^3), of air.

Table 2.2.1: Key conversions between relative ppm-values of CO_2 to absolute volumes.

Relative values		Absolute values given ½ liters of air	
ppm	Volumetric-%	ml	dm^3
400	0.04	0.2	0.0002
1000	0.1	0.5	0.0005
40000	4	20	0.02
50000	5	25	0.025

2.2.2 Diaphragmatic Breathing

The principle of diaphragmatic breathing proceeds in the following manner: Air is inhaled when the diaphragm contracts, expanding the lung sac volume and causing a negative pressure gradient to appear within the system. Air is then exhaled when the diaphragm is relaxed, causing the lung sacs volume to shrink and a positive pressure gradient to appear within the system [7]. Figure 2.2.1 visualises the principle of diaphragmatic breathing.

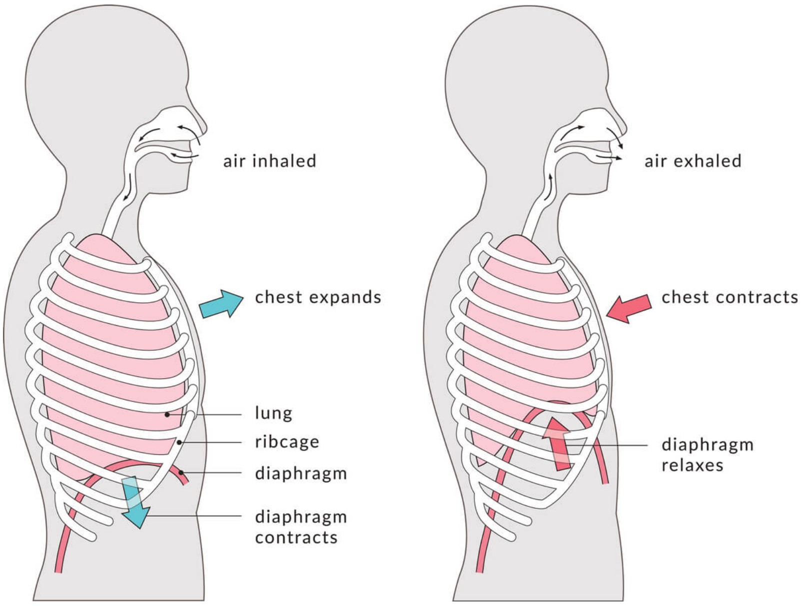


Figure 2.2.1: A demonstration of diaphragmatic breathing [9].

2.3 Carbon Dioxide Release

In order to control the release of CO_2 information about the dynamics of the flow has been gathered together with information about the components that is going to be used to control the release. The precision of a CO_2 release system will depend on how small the pressure gradient between the CO_2 gas and the ambient air can be made and how rapidly the outlet can open and close itself. The release of CO_2 will be controlled by pressure regulator together with a solenoid valve [10].

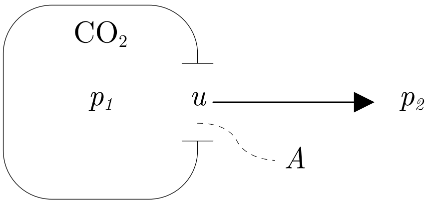


Figure 2.3.1: A CO_2 container and an outlet with the cross-section area A. The gas moves from the high pressure environment p_1 to the low pressure environment p_2 with the velocity u.

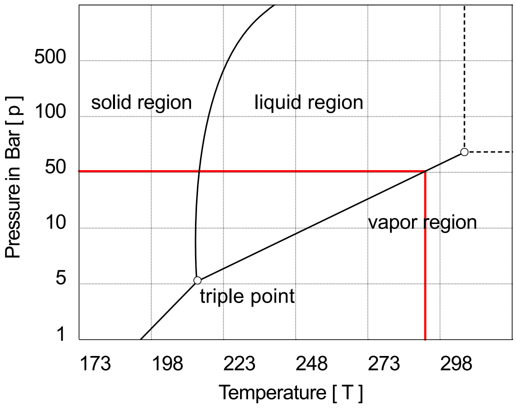


Figure 2.3.2: A pT-diagram of CO_2 with the liquid-vapor boundary at room temperature highlighted in red [11].

CO_2 containers are highly pressurised with most of the CO_2 in a liquid state. The vapor above the liquid-vapor boundary has a pressure of around 50 bar (see Figure 2.3.2) assuming that the container is in room temperature, here defined as 293.15 Kelvin (K) or 20 Degrees Celsius (°C). This is about 50 times higher than the ambient pressure at sea level, 1.013 bar [12].

2.3.1 Bernoulli's Principle

Bernoulli's principle states that the total pressure, p_0, along a streamline is constant. The total pressure can further be said to be the sum of the static pressure, p, and the dynamic pressure, $\rho u^2/2$, neglecting any major difference in elevation along the streamline [10]:

$$p_0 = p + \frac{\rho u^2}{2}. \tag{2.1}$$

If the gas inside the container in Figure 2.3.1 is assumed to be stationary (on average), equation 2.1 can be rewritten as

$$u = \sqrt{\frac{2 \cdot (p_1 - p_2)}{\rho}}. \tag{2.2}$$

The volume flow, $\dot{V}$, is the product of the cross-section area of the outlet-opening, A, and the gas velocity described in equation 2.2:

$$\dot{V} = uA. \tag{2.3}$$

If the outlet is open for a set period of time, ranging from $T_1 = 0$ to $T_2 = t$, the final volume, V, of the released gas is

$$V = \int_0^t \dot{V} dt = At\sqrt{\frac{2 \cdot (p_1 - p_2)}{\rho}}. \tag{2.4}$$

It is from equation 2.4 evident that the key factors that decide the volume of released gas are:

1. The cross-section area of the outlet, A.
2. The pressure gradient, $p_1 - p_2$.
3. The duration of the outlets open-state, t.

2.3.2 Pressure Regulator

A pressure regulator can be positioned between a pressure source and a device or outlet to either reduce or increase the pressure. Additionally a pressure regulator will work to maintain a constant pressure, even if the pressure source experiences variations in its pressure. The basic principle of a pressure regulator is that it makes use of a primary and secondary chamber with a closing mechanism between them. The primary chamber is connected to the pressure source while the secondary chamber is connected to the outlet of the regulator. The closing mechanism is constructed in such a way that it stays closed when there is a force-balance between the forces acting upon it from the primary and secondary chambers. The force in the primary chamber is solely from the pressure itself while the force in the secondary chamber is both caused by the gas pressure and a static force created by an adjustable spring. This lets the closing mechanism remain shut even though there is an imbalance between the pressures of the medium flowing through the regulator [13].

2.3.3 Solenoid Valve

A solenoid valve is an electro mechanical valve which uses an electromagnetic force to open or close itself. The valve contains a coil which, when current runs through it, induces a magnetic field. This magnetic field acts upon a spring-loaded piston and, if the magnetic field is strong enough, overcomes the spring-force thus opening or closing the valve. This way an electrical signal can be translated into pneumatic or hydraulic functions [14]. The valve can earthier be Normally closed (NC) or Normally open (NO). This report will focus on the NC solenoid valve meaning that the valve remains closed when no current is applied to it [14].

2.4 Control theory

A feedback controller can be used on order to get a system to behave as desired. The principal of a feedback controller is based on that the system has a desired input signal which is given to the controller, which gives an input signal to the system and the system gives the resulting output signal. This output signal can then be used to compare the reference signal. The controller's adjustments of the new input signal into the system is based on that comparison. In Figure 2.4.1 a schematic of a feedback loop is shown where r represents the reference signal, the e represents the difference between the reference signal and the actual resulting output form the system, y which was a result from the input signal u from the controller into the system [15].

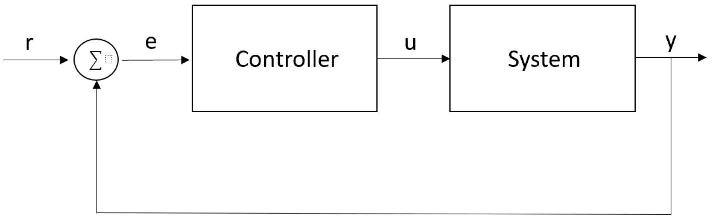


Figure 2.4.1: Feedback Loop.

To evaluate a designed control system an analysis of the step response can be done. Interesting data points from the step response could be the settling time and rise time. Rise time can be determined by looking at the time it takes for the system to go from 1% of the final value to 90% of the final value. The settling time is defined as the time it takes for the system to reach and stay within 5% of the desired value [15].

Chapter 3

Demonstrator

The goal for the system that has been constructed is to simulate the human respiratory system during sleep. It will do so by pumping in ambient air into an enclosure and adjust the CO_2 saturation and humidity of the air to resemble the levels found in the exhaled air of an average adult human while resting. Once the manipulations have been made the air will be pumped out of the enclosure to complete the cycle. The system can be broken down into three codependent subsystems; the pumping mechanism, the CO_2-system, and the humidity system. A flowchart describing how the different systems integrates with each other can be seen in Figure 3.0.1.

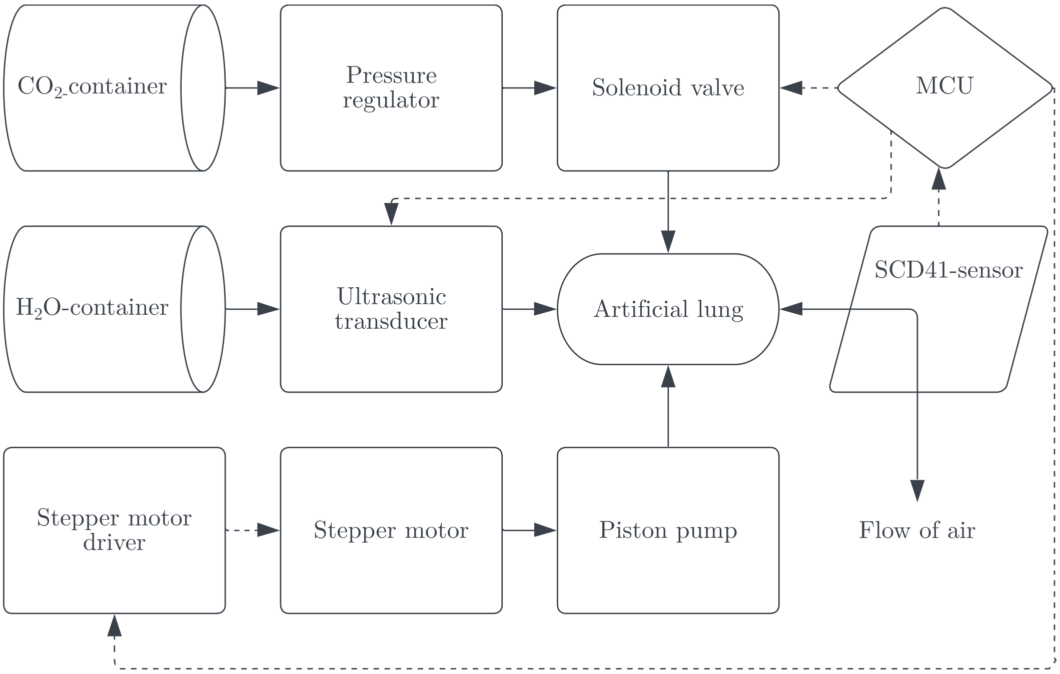


Figure 3.0.1: Flow chart of the various sub-systems of the machine. Solid lines denote physical interaction (flow of media or motion) and dashes lines denote signal-flow.

3.1 Electronics

This section will describe the different electronic components that were required for the machine to operate as desired. Figure 3.1.1 demonstrates the circuit connection between the components, all the electronic components are centered around the microcontroller.

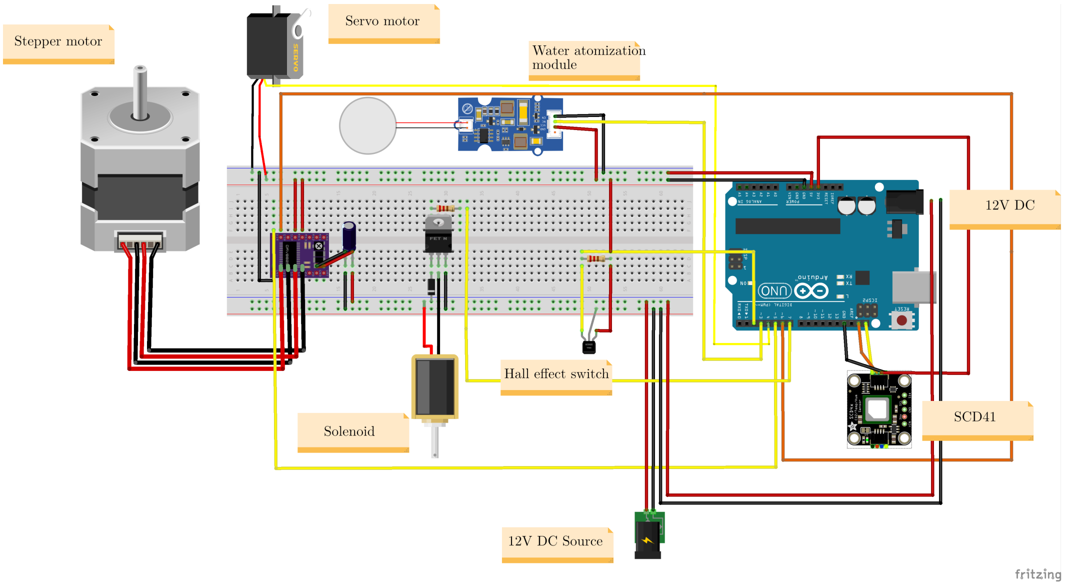


Figure 3.1.1: Electrical circuit for the system.

3.1.1 Microcontroller

An Arduino UNO microcontroller was used to control the components. The Arduino is compatible with its own integrated software based on the programming language C++. It has 14 digital input/outputs and 6 analog inputs.

3.1.2 Hall effect sensor

To control the positioning of the piston a hall effect sensor were used [16] and can be seen in Figure 3.1.2. The sensor recognizes magnetic fields and changes the output voltage after a change in the magnetic field is felt.

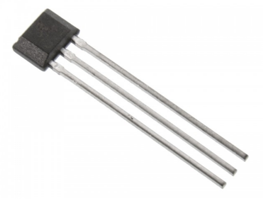


Figure 3.1.2: TLV4964-5T Hall effect switch [16].

3.1.3 CO_2- temperature- and humidity sensor

The SCD41 sensor from Sensirion was used to measure the temperature, humidity and CO_2 in the system [17]. The sensor can be seen in Figure 3.1.3 The sensor measures CO_2 levels between 0 ppm to 40 000 ppm with a typical response time of 60 seconds. The sensors also has an integrated temperature and humidity sensing element. It measures humidity in the range of 0 % RH to 100 % RH and temperature between -10 °C to 60 °C. The required supply voltage range between 2.4 Volts (V) to 5.5 V.

Figure 3.1.3: SCD41 Sensor [17].

3.1.4 Water atomization module

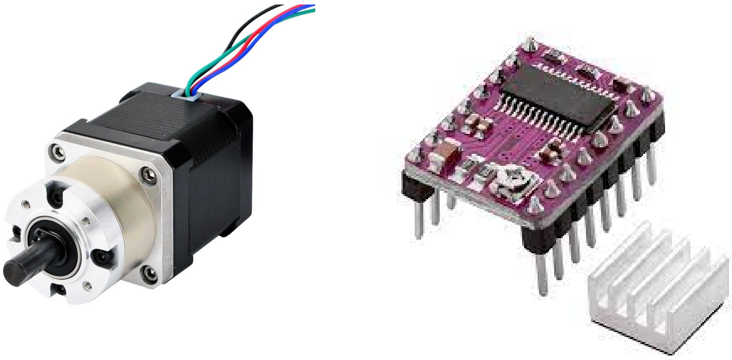
A water atomization module was used to supply the system with humidity. The module can be seen in Figure 3.1.4 and heats water with ultrasound which creates humidity. It has an operating voltage of 5 V [18].

Figure 3.1.4: Grove - Water Atomization [18].

3.1.5 Motors

Stepper motor

A Nema 17 stepper motor was used to operate the pumping mechanism [19]. The motor uses the DRV8825 motor driver carrier as connection [20]. The driver together with the motor can be seen in Figure 3.1.5. The driver have different micro-step resolution and operates from 8.5 V to 45 V.



((a)) Nema 17 Stepper motor [19]

((b)) DRV8825 Driver [20].

Figure 3.1.5: Stepper motor with motor driver.

Servo motor

An SG90 servo motor was used to operate the closing- and opening mechanism that worked to partially seal the inlet/outlet of the artificial lung while the contained air was being manipulated by the system [21].

Figure 3.1.6: sg90 Servo motor [21].

3.1.6 Solenoid valve

A solenoid valve was used for controlling the amount of CO_2 that was let into the system [22]. The normally closed valve was connected with a transistor letting it run with 12 V. Figure 3.1.7 shows the solenoid valve.

Figure 3.1.7: Solenoid valve [22].

3.2 Software

The microcontroller's software program was constructed with a setup section followed by a calibrations section where the piston were placed in the right position. It was followed by the main program which drives the motor for the pumping mechanism and preforms the manipulation of the air according to the desired values of CO_2 levels and humidity. There is also a re-calibration step of the piston which takes place every 10 minute. Figure 3.2.1 demonstrates the general process during normal operation.

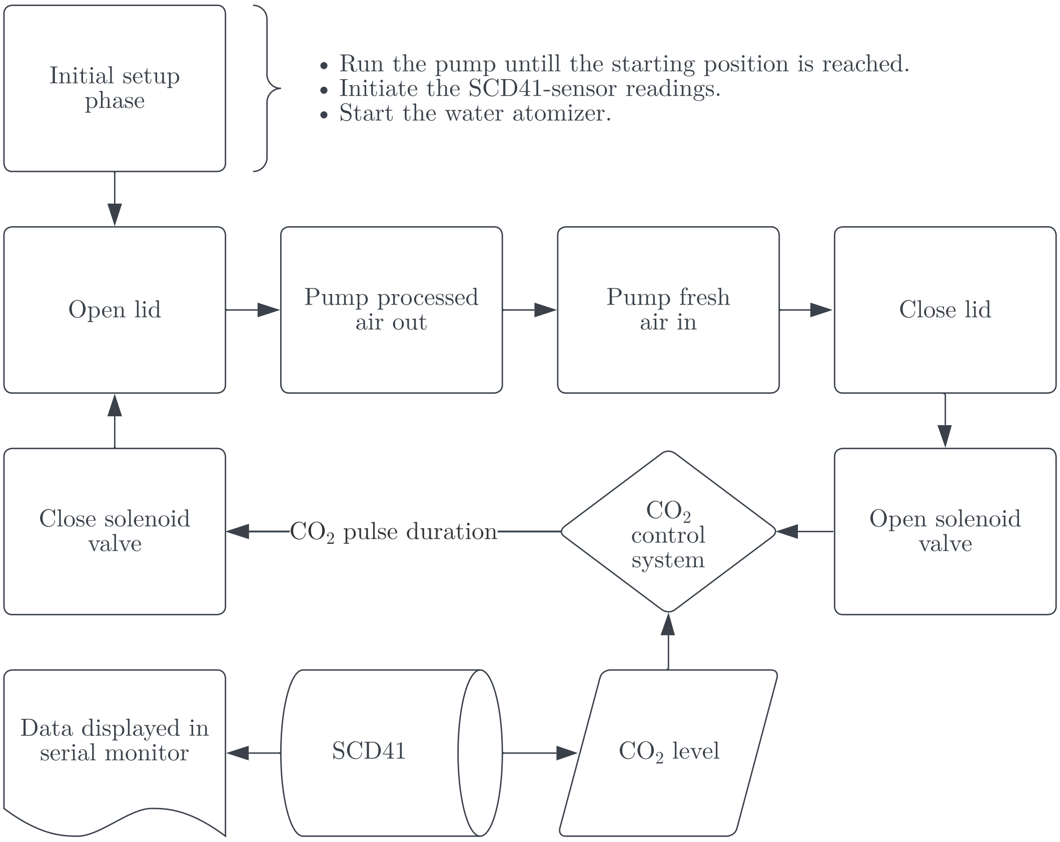


Figure 3.2.1: Flowchart of the microcontroller runtime.

For simplicity (and because of the slow response time of the SCD41 sensor) the CO_2 control system (a feedback loop controller) was written in such a way that the steering signal (the time delay between the opening and the closing of the solenoid valve) only gets to update itself to a new value once the sensor reports a steady signal (when the derivative of the error between the reference value and the feedback value is low). This ultimately creates a very slow step response but should ensure that the system approaches a steady state close to the set reference value without too much oscillation. To further avoid system instability artificial boundaries were set in the code to prevent the steering signal to reach unreasonable values. Finally, to give the system a head start an ”initial guess” was constructed in the way of a linear equation that approximates the empirically measured steady state values for various control signals. How the software is implemented can be seen in Appendix A.

3.3 Calibration of the SCD41 sensor

The SCD41 sensor comes with a couple of calibration methods built into its system, Forced recalibration (FRC) and Automatic self-calibration (ASC). To use FRC there is a need for a gas of precisely known CO_2 concentration. The simpler ASC calibration, which was applied, needs no additional equipment by the end user when enabled. The only requirement for ASC to work is that the sensor has to be exposed to atmospheric levels of CO_2 at least once per week [17].

To perform a sanity check on the sensor, it was left running in both a crowded room where it measured a CO_2 level of 886 ppm and inside of a plastic bag which had been filled with exhaled air where it reported a level of 34572 ppm. Both of these measurements aligned with what is to be expected. However, no external CO_2 sensor was used to verify the claims of the SCD41 sensor.

3.4 Design

A general design of the machine can be seen in Figure 3.4.1 and 3.4.2. The different systems are constructed inside the enclosure which was assembled with acrylic plates. Inside the larger enclosure a smaller enclosure is placed which will be working as the ”lung”. An opening is placed on one of the sides of the big enclosure which enables the inhaled air to be transported into the smaller enclosure and then released again once processed.

Figure 3.4.1: An overview of the design.

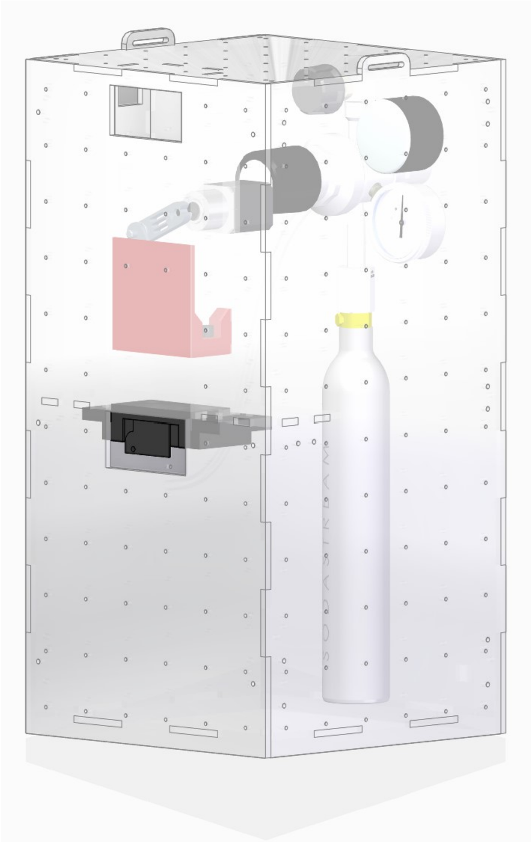


Figure 3.4.2: An overview of the design, made in Solid Edge.

3.4.1 Pumping mechanism

To closely resemble the mechanism that is responsible for pumping air in and out of a human lung, the diaphragm, a hermetically sealed (in all directions except the in- and out-take) piston pump is suitable in its simplicity. The model is shown in figure 3.4.3 and the construction is located inside the ”lung” and can be seen in figure 3.4.2. The enclosure under the piston has been chosen to be demonstrated with a plastic bag with the purpose to prevent leakage of the air and also in which the air manipulations will take place.

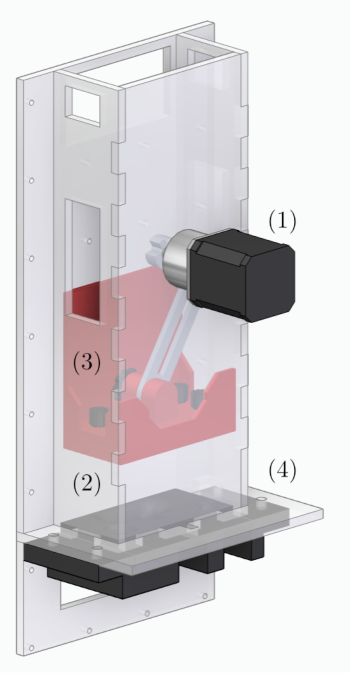


Figure 3.4.3: Model of pumping mechanism. (1) Stepper motor. (2) Airbox (artificial lung). (3) Piston assembly. (4) Bottom plate with inlets for the solenoid valve, the water atomizer, and the SCD41-sensor.

On top of the piston a crank arm is attached which is connected to a stepper motor enabling it to move in a breathing motion. To verify that the breathing cycle starts in the right position a neodymium magnet is placed on top of the crank arm which will be within the range of a hall effect sensor at the top of the stroke. On one of the walls of the enclosure where the pump is mounted, a magnet is located which can be detected by the sensor and inform the system about the position of the crank arm.

The average adult human breath contains roughly half a liter (or 0.5 dm^3) of air, this constricts the stroke volume of the piston pump (the bore area multiplied by the length of the stroke) [4]. To ensure good circulation, meaning that most of the processed air is pumped out with each cycle, the total volume should be kept as close to the stroke volume as possible.

On average an adult human breathes somewhere between 12 and 20 times per minute while resting, this translates to a piston pump that completes one cycle every three to five seconds [4]. If a three second cycle is chosen there is room for a two second pause when the piston pump is at its maximum volume state. During this two second temporal window the other sub-systems should have time to make sure that the different parameters of the enclosed air that are to be controlled are within specification before the piston pump proceeds to exhale the processed air.

Since CO_2 is denser than atmospheric air [23][24] the CO_2-enriched air will tend to migrate downwards if no other forces act upon it. For this reason, it is beneficial to have the piston pump oriented in such a way that the in- and out-take is at the bottom.

3.4.2 CO_2 system

In Figure 3.4.4 a model of the CO_2 control system can be seen. It is controlled by a pressure regulator together with a solenoid valve. The source of the CO_2 is a carbonation machine CO_2 bottle with a pressure of 50 bar [25]. To be able to work with such a highly pressurised gas a pressure regulator has been used for the system. The regulator can be used to lower the gas from 50 bar to below 0.5 bar of delta pressure, where delta pressure is the relative pressure compared to the atmospheric pressure outside of the system. A secondary benefit of such a regulator is that the output-pressure will remain constant even if the input-pressure (that inside of the CO_2-container) would lower as the CO_2 is depleted.

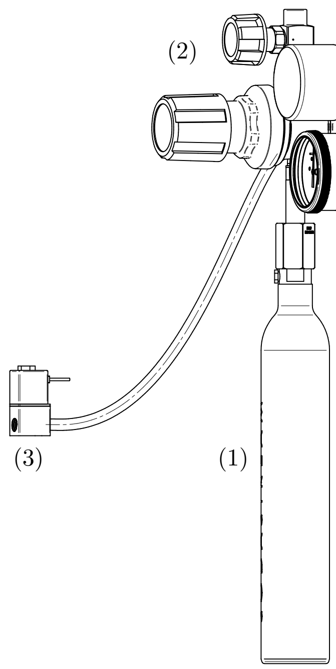


Figure 3.4.4: Model of the CO_2 system. (1) is the CO_2-container, (2) is the pressure regulator, and (3) is the solenoid valve. The model was made in Solid Edge.

Once the pressure of the CO_2 has been regulated down to a manageable and predictable level a normally closed solenoid valve is used to let out bursts of CO_2-gas in a controlled manner. By leaving the valve open for a longer or shorter duration combined with the knowable factors such as the exit-velocity of the gas and the opening-area of the solenoid valve, a sought volume can be translated into a sought opening-duration for the solenoid valve.

A tube is connected between the CO_2 bottle and the solenoid valve and is placed in the bottom of the smaller enclosure where the air is being modified.

3.4.3 Humidity system

An automated water atomizing module (ultrasonic transducer) is used to add water vapor into the air. It is located on the bottom of the smaller enclosure and is attached to a sponge that has been soaked in water.

Chapter 4

Experiments

This chapter will present the experiments that was conducted in the project for the purpose of answering the questions presented in Section 1.2.

4.1 Solenoid valve speed test

To evaluate the limits of the solenoid valve, an experiment was devised where a pulse (the valve opens and closes again during a specified time duration) of CO_2 at some pressure (preferably low to increase the resolution of the test) was injected into a plastic bag. The plastic bag was then lowered into a body of water contained inside a container with a known cross-section area (A). The difference between the water-level before and after the bag was lowered into the water (h_0 and h_1) was then measured, and the volume of water that the plastic bag displaced could be calculated as follows:

$$V_{CO2} = V_{displaced\ water} = A \cdot (h_1 - h_0). \tag{4.1}$$

The initial pulse time duration was then divided by two and then used to inject CO_2 into the plastic bag twice (meaning that a one second initial pulse of CO_2 was turned into two half-second pulses). This pattern (halving the pulse time duration and doubling the number of pulses) was repeated until the total volume of CO_2 that the system produced either changed significantly or the solenoid valve showed other signs of failure (either not opening or not closing properly for example). The limit (in terms of minimum pulse time duration capacity) of the solenoid valve could then be decided by looking at the lowest pulse time duration that showed signs of stability.

4.2 Steady state levels

The complete CO_2-control system (the pump, the CO_2-injection subsystem and the CO_2-sensor) is relatively slow, mostly because of the relatively slow response time of the CO_2-sensor ($\tau_{63} = 60$ seconds [17]). To speed the system up the control system can be given a reasonable initial guess for the pulse time duration needed to achieve a specific CO_2 output.

To be able to formulate an equation for the initial guess the systems combined characteristics were evaluated through a series of test-runs where a fixed time duration for the CO_2-injection pulse was used for a long time until the CO_2-sensor reported a steady state value. The system was then ventilated and restarted using a different fixed time duration for the CO_2-injection pulse. These 2-tuples of data (the pulse time duration and the steady state value achieved) could then be plotted in a graph and curve-fitting could be applied to formulate an equation for the initial guess.

Finally a series of test runs with the pulse time duration set to the same value was performed to evaluate the systems repeatability.

4.3 Finalised system evaluation

To evaluate the calibrated systems performance a series of test runs were made. During the test runs the machine was given different target CO_2 values which it was supposed to rise to (from the background-level) and then maintain for a period of time to display stability. The test runs were plotted on CO_2-time plots and the performance was evaluated based on rise-time, settling time and the steady states peak deviation from the reference value.

A test-run with the water atomizer running was also performed, to evaluate the performance of the humidifier within the system.

Chapter 5

Results

In this chapter the results from Section 4 will be presented.

5.1 Solenoid valve speed test

Displayed in table 5.1.1 and in figure 5.1.1 are the gathered data from the solenoid valve speed test explained in section 4.1.

Table 5.1.1: A table displaying the two attempts from each pulse time duration.

Pulse time duration (ms)	*Number of pulses*	*Attempt 1 (ml)*	*Attempt 2 (ml)*
2000	1	308	284
500	4	272	296
250	8	296	319
125	16	308	284
62,5	32	260	272
31,25	64	296	272
15,625	128	284	248
7,8125	256	237	284
3,90625	512	130	24
1,953125	1024	0	0

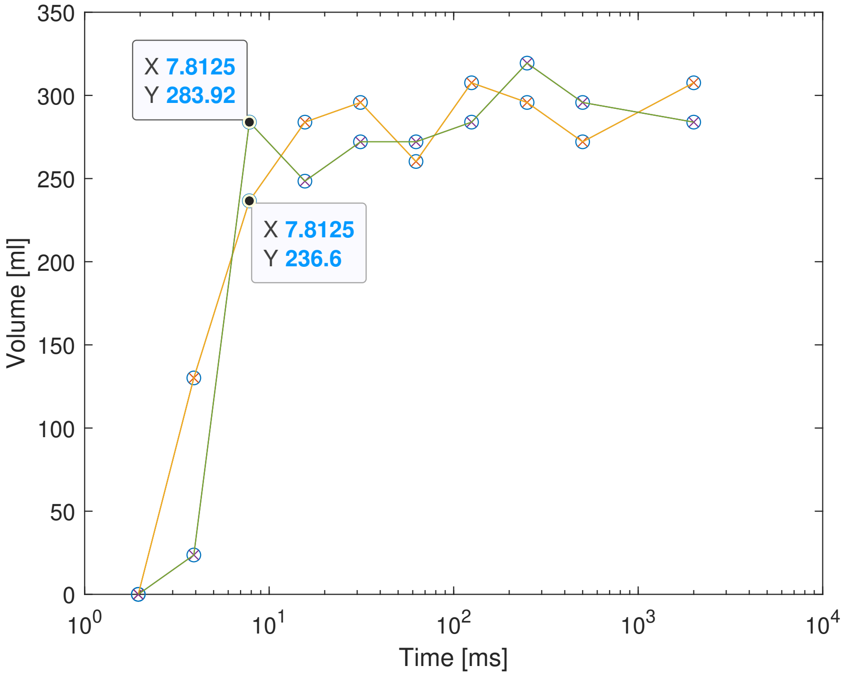


Figure 5.1.1: The blue line is from the first attempt and the orange line is from the second attempt. The time-axis is reversed and logarithmic.

5.2 Steady state levels

Figure 5.2.1 and Figure 5.2.2 together with Table 5.2.1 shows results from the experiments explained in Section 4.2.

Table 5.2.1: A summary of the average steady state values reported by the SCD41-sensor during test-runs.

Pulse time duration (ms)	*CO2 steady state value (ppm)*
10	5 500
20	9 700
30	13 400
40	17 400
50	20 500
60	23 500
70	27 500
80	30 000

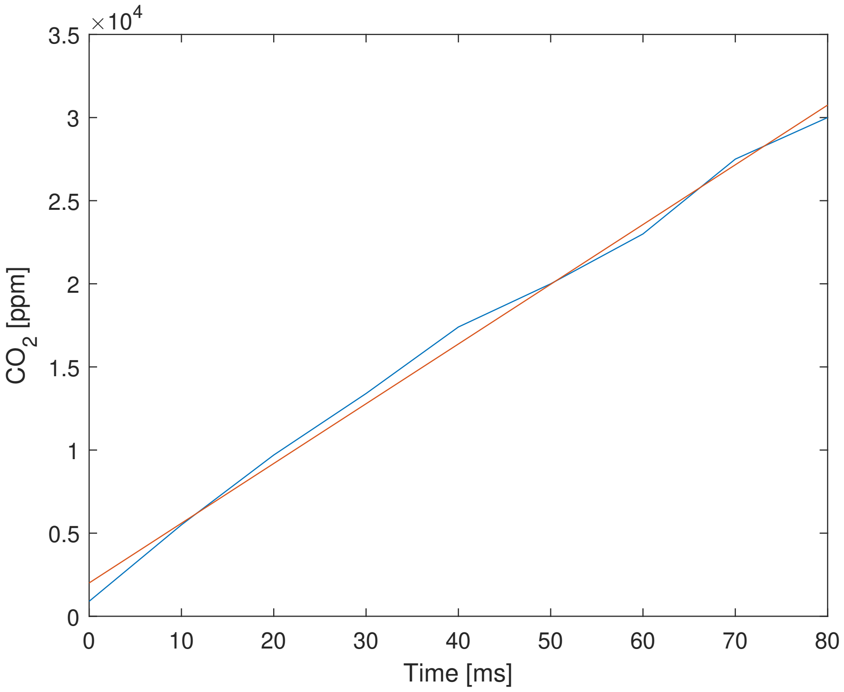


Figure 5.2.1: The blue line is the measured steady state levels for all the tested pulse time durations. The orange line is the simple linear regression model with the equation $y_{CO_2} = 359 \cdot t + 2004$.

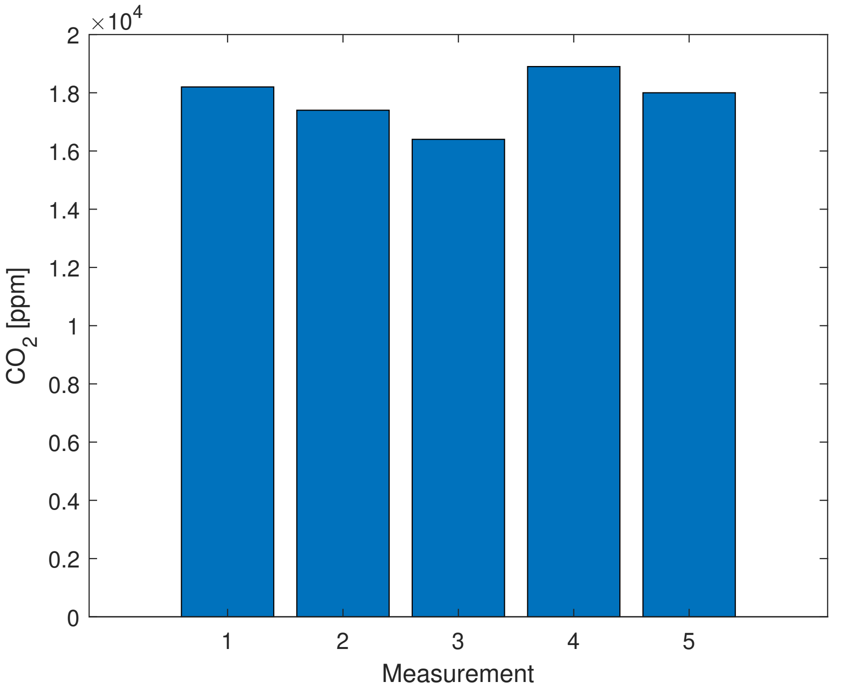


Figure 5.2.2: The variation between test runs with a pulse time duration of 40 ms.

5.3 Finalised system evaluation

Figure 5.3.1, Figure 5.3.2 and Figure 5.2.2 shows the results from the experiments explained in Section 4.3.

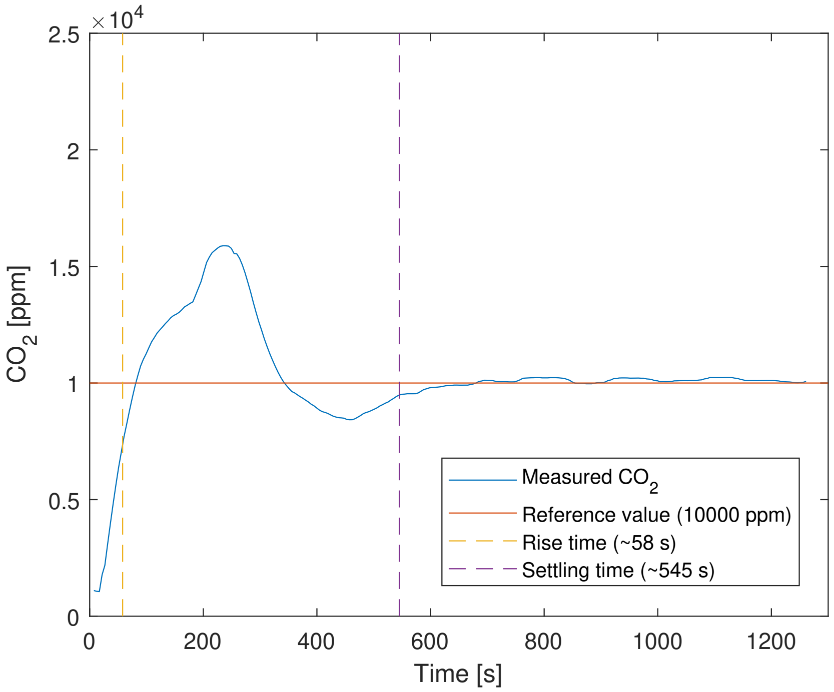


Figure 5.3.1: The step response curve for the system when the reference value was set to 10 000 ppm.

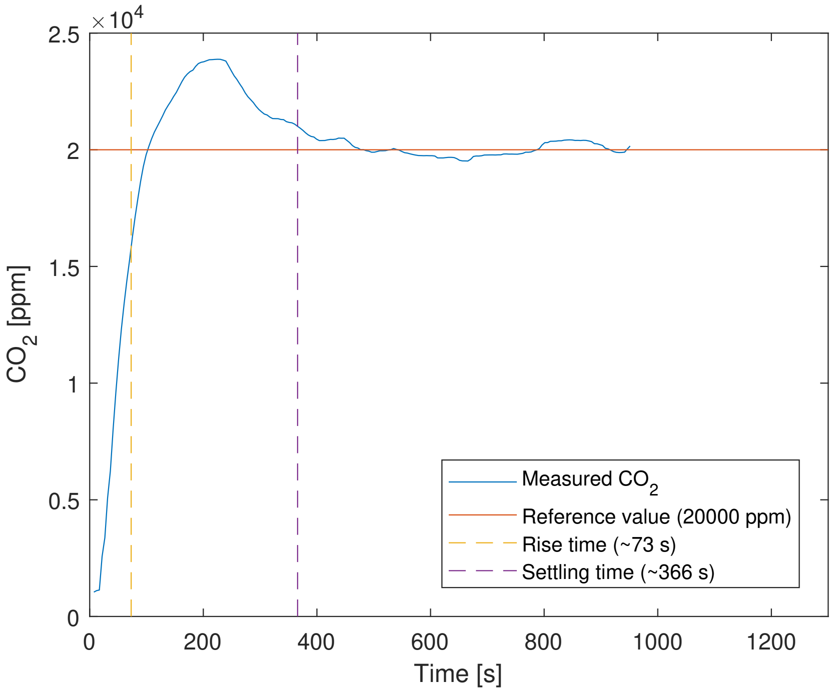


Figure 5.3.2: The step response curve for the system when the reference value was set to 20 000 ppm.

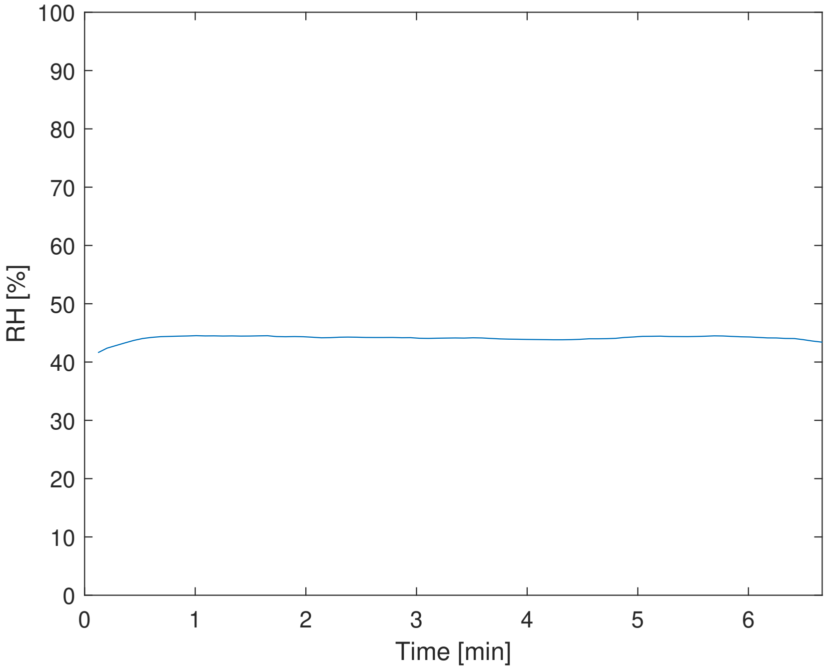


Figure 5.3.3: RH plotted over a runtime of 400 seconds.

Chapter 6

Discussion and Conclusions

6.1 Solenoid valve speed test

Looking at table 5.1.1 and the accompanying graph in Figure 5.1.1 it seems like a pulse time duration of roughly 8 ms is at the limit of what the solenoid valve can manage with somewhat predictable outputs. Any attempts to go lower resulted in huge variations of the final CO_2 output or the failure of the solenoid valve to open whatsoever.

The relatively large fluctuations along the span of pulse time duration that are longer than 8 ms could either be due to the relatively crude measuring equipment used to determine the water-level (a ruler) or due to actual dynamic variations within the CO_2-release system. A more well-prepared test with higher measuring accuracy and a larger number of trials could be used to rule one or the other out.

6.2 Steady state levels

The steady state for the pulse time duration ranging from 10 ms to 80 ms (in 10 ms steps) displayed linear characteristics of the system (the blue line in Figure 5.2.1, see table 5.2.1 for the numeric values). This simplified the choice of an equation (a linear one, as displayed by the orange line in Figure 5.2.1) to base the initial guess for the pulse time duration upon.

The repeatability-test at the mid-point, 40 ms, had a maximum measured output of 18 900 ppm and a minimum measured output of 16 400 ppm, as seen in Figure 5.2.1.

As close as the chosen linear equation (in Figure 5.2.1) matches the measured steady state values for the various pulse time durations is, it should still only be treated as a very crude initial guess to get the system close to the final value as soon as possible. Small deviations in the CO_2-pressure or a failure of the artificial lung to reach its full potential volume (the plastic bag crumpling or a missed step by the stepper motor) could drastically affect the needed slope of the linear equation (as evident from the repeatability-test). Furthermore any dramatic changes to the environments background levels of CO_2 would cause a deviation in where the linear equation should intercept the "y-axis".

6.3 Finalised system evaluation

The finalised systems test-runs, show in Figure 5.3.1 and Figure 5.3.2, showed promising signs of stability in regards of the CO_2 output. The system is, as expected, slow (with a settling time of just over 9 minutes for the 10 000 ppm test-run). A more robust and well-written controller would likely accelerate the process. However, as the machine is intended to run for many hours, even a settling time approaching 10 minutes is acceptable and the result of interest was always the ability to actually reach and maintain a steady state at the set reference value, which it managed to do in all of the test-runs.

The humidifier did not achieve the sought after RH-level of 100 %. From the test-run data, seen in Figure 5.3.3, it is hard to evaluate if it managed to increase the RH-level at all, as the initial deviations are within the ± 6 % error-bars [17].

6.4 General discussion about the system

The design of the machine can be evaluated from its performance during the experiments in section 4. The machines design has shown to be simple and robust and able to perform subsequently breathing motions over a longer period of time. The machines design together with its components are also able to self regulate its position in the breathing cycle which is crucial for the CO_2 control system. It is possible to conclude that the general design of the prototype for the machine is suitable for a machine aiming to resemble human breathing based on diaphragmatic breathing. However, the machine has some weaknesses including its ability to prevent leakages of air which likely causes faults in the collected data.

Bedbugs are highly dependent on regular blood meals to be able to survive and reproduce, this causing a lot of stress for individuals and damages to different industries. To treat an infestation today a human is needed to be present in the room mainly to act as a bait for the bedbugs. As previous research has shown that increased levels of CO_2 is the main attractant for bedbugs this machine should be able to act as substitute. The machine has the possibilities to reach and maintain a steady value of desired CO_2 and can also reach values as high as the humans CO_2 levels of exhaled air.

6.5 Conclusions

- **How could a machine be designed to resemble human breathing?**
  It was found that while the CO_2 release sub-system performed well, the humidity system has to be redesigned using other components and another approach to be able to reach and maintain a level of humidity found in human breath.

  The breathing apparatus could be simulated sufficiently using a piston-pump design, however using a plastic bag to prevent leakage caused a new set of problems, an undesirably variable internal volume primarily.

- **How can a control system be designed to ensure that the machine can reach and maintain a desired level of CO_2 release?**
  Through the results of this paper it has been possible to conclude that even a simple control system and basic components are sufficient to reach and maintain a desired CO_2 level for a longer period of time, as fluctuations of

the CO_2 levels outside of the systems rarely exceed a couple of hundred ppm which is a tiny amount compared to the tens of thousands ppm that the machine produces during operation.

- **How could a CO_2 release system be designed to resemble CO_2 release similar to the human respiratory system?**
  While having the aforementioned flaws, a simple piston-pump in which CO_2 was injected in tiny doses into the piston-pumps internal volume before the piston-pump pressed the air out proved to be very effective. However, no measurements were ever performed outside of the machines internal volume which leaves the question partially unanswered regarding its similarity to the human release of CO_2.

Chapter 7

Further Work

- **Try the machine on bedbugs**
  To evaluate if the machine does attract bedbugs and can be a substitute for a human during treatment for an ongoing infestation, suitable tests on bedbugs has to be made and evaluated.
- **Try other bedbug attractants**
  It is also necessary to investigate other bedbug attractants for the machine to emit to increase the chances of successfully attract bedbugs.
- **Longer test evaluations**
  The machine has been running for up to one hour but longer duration is necessary to fully evaluate if the machines design could fulfil the need for the machine to be running up to six hours, several nights in a row.
- **Trials with external sensor**
  No external sensor was used during this project which means that the machines ability to simulate (in terms of CO_2-distribution) human presence in a room still remains unanswered. Trials with an external sensor would be beneficial to further evaluate the system.
- **Air leakage**
  As mention in section 6.4 the machine has a weaknesses of preventing leakages of air, this could be improved with another design of the enclosure where the air manipulation takes place.
- **Sensor**
  A sensor with a faster response time would improve the CO2 control system.
- **Humidity system**
  As mention in section 6.3 the design for the humidity system was not suitable for the machines design. This part has to be improved in order to have added humidity in the simulation.

Bibliography

[1] Reinhardt, Klaus and Siva-Jothy, Michael T. “Biology of the Bed Bugs (Cimicidae)”. In: *Annual Review of Entomology* (2007). DOI: `10.1146/annurev.ento.52.040306.133913`. URL: `https://www.annualreviews.org/doi/pdf/10.1146/annurev.ento.52.040306.133913`.

[2] Anderson, John F., Ferrandino, Francis J., and al., Michael P. Vasil et. “Relatively Small Quantities of CO2, Ammonium Bicarbonate, and a Blend of (E)-2-Hexenal Plus (E)-2-Octenal Attract Bed Bugs (Hemiptera: Cimicidae)”. In: *Annual Review of Entomology* (2017). DOI: `10.1093/jme/tjw189`. URL: `https://academic.oup.com/jme/article/54/2/362/2970131`.

[3] “Stick.se”. In: *Website* (-). URL: `https://stick.se/blogg/sanering-vaggloss/`.

[4] GmbH, WILAmed. “Respiratory Humidification: Basics”. In: *Information sheet* (2012). URL: `https://www.vingmed.se/wp-content/uploads/2014/03/Broschyr_EN.pdf`.

[5] Miller, Dini M., Polanco, Andrea, and Rogers, Jeffrey. “Bed Bug Biology and Behavior”. In: *Virgina Cooperative Extension* (2013). URL: `https://www.vdh.virginia.gov/content/uploads/sites/119/2018/02/Bed-Bug-Biology.pdf`.

[6] Wang, Changlu, Gibb, Timothy, Bennett, Gary W., and McKnight, Susan. “Bed Bug (Heteroptera: Cimicidae) Attraction to Pitfall Traps Baited With Carbon Dioxide, Heat, and Chemical Lure”. In: *Journal of Economic Entomology, Volume 102, Issue 4* (2009). DOI: `10.1603/029.102.0423`. URL: `https://doi.org/10.1603/029.102.0423`.

[7] Das, Sagnik and Pal, Mrinal. “Review—Non-Invasive Monitoring of Human Health by Exhaled Breath Analysis: A Comprehensive Review”. In: *Journal of The Electrochemical Society* (2020). DOI: `10.1149/1945-7111/ab67a6]`. URL: `https://iopscience.iop.org/article/10.1149/1945-7111/ab67a6/pdf`.

[8] Arshad, F. “A Study of Smart Ventilation System for Maintaining Healthy Living by Optimal Energy Consumption: A case study on Dalarnas Villa”. In: (2020). URL: `https://www.diva-portal.org/smash/get/diva2:1525233/FULLTEXT01.pdf`.

[9] VipCare. “Proper Breathing: How You Might Be Doing It Wrong”. In: *Website* (2020). URL: `https://getvipcare.com/blog/proper-breathing-how-you-might-be-doing-it-wrong/`.

[10] Karlsson, Arne. “Formelsamling i Strömningsmekanik: Upplaga 2.11”. In: *Book* (2007).

[11] Gas, Linde. “Safety advice. Carbon Dioxide.” In: *Information sheet* (2017). URL: `https://www.linde-gas.com/en/images/LMB_Safety%20Advice_01_66881_tcm17-165650.pdf`.

[12] SMHI. “Air pressure and sea level”. In: *Website* (2014). URL: `https://www.smhi.se/en/theme/air-pressure-and-sea-level-1.12266`.

[13] Skousen, Philip L. “Valve Handbook”. In: (1997). URL: `http://dl.mohandes-iran.com/mekanik/HANDBOOK/Valve%20Handbook.pdf`.

[14] Steinsland, Vidar. “Modeling and Control of Retarder using On/Off Solenoid Valves”. In: *Independent thesis Advanced level* (2008). URL: `http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A572695&dswid=865`.

[15] Glad, Torkel and Ljung, Lennart. “Reglerteknik Grundläggande teori”. In: *Book* (2006).

[16] “TLV4964-5T”. In: *Datasheet* (2011). URL: `https://www.electrokit.com/uploads/productfile/41015/2343219.pdf`.

[17] “SCD4x”. In: *Datasheet* (2021). URL: `https://sensirion.com/media/documents/C4B87CE6/61652F80/Sensirion_CO2_Sensors_SCD4x_Datasheet.pdf`.

[18] “Grove - Water Atomization”. In: *Datasheet* (2012). URL: `https://wiki.seeedstudio.com/Grove-Water_Atomization/`.

[19] “17 Nema Stepper motor”. In: *Datasheet* (Unknown year). URL: `https://www.omc-stepperonline.com/download/17HS19-1684S-PG5.pdf`.

[20] “DRV8825 Motor driver”. In: *Datasheet* (2014). URL: `https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1650454998431&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FDRV8825%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-mdbu-null-prodfolderdynamic-cpc-pf-google-wwe%2526utm_content%253Dprodfolddynamic%2526ds_k%253DDYNAMIC%2BSEARCH%2BADS%2526DCM%253Dyes%2526gclsrc%253Daw.ds%2526gclid%253DCj0KCQjw3v6SBhCsARIsACyrRAk-vjO-N3xuUzVR6Qv-vTaw-DHf2ZoEhU4DUZLwsef8YvN1iyRouCoaApEiEALw_wcB`.

[21] “SERVO MOTOR SG90 ”. In: *Datasheet* (Unknown year). URL: `http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf`.

[22] “Solenoid valve”. In: *Website* (2021). URL: `https://www.amazon.se/Solenoid-luftventil-pneumatisk-aluminium-magnetventil/dp/B07STBDP7C/?fbclid=IwAR2OcMbGi3SA6xjaYClQhzb2n4odF5UEDcFCdqENO5uotZaijBSzMDZs_2w`.

[23] ToolBox, Engineering. “Carbon dioxide - Density and Specific Weight vs. Temperature and Pressure.” In: (2018). URL: `https://www.engineeringtoolbox.com/carbon-dioxide-density-specific-weight-temperature-pressure-d_2018.html`.

[24] ToolBox, Engineering. “Air - Density, Specific Weight and Thermal Expansion Coefficient vs. Temperature and Pressure.” In: *Website* (2003). URL: `https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html`.

[25] “CO2 container”. In: *Datasheet* (2017). URL: `https://www.clasohlson.com/medias/sys_master/9543017955358.pdf`.

Appendix - Contents

Appendix A

Appendix

A.1 Arduino code

```
// IFIGENIA ARDUINO-KOD \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
//
// Skapad av:         Maja  strand  (jastran@kth.se) & Simon Lilja (simlil@kth.se)
// Senast modifierad: 2022-05-06
// \\\\\\\\\\\\\\\\\\\\\\\\\


// BIBLIOTEK \\\\\\\\\\\\\\\\\
#include <Arduino.h>
#include <SensirionI2CScd4x.h>
          SensirionI2CScd4x scd4x;
#include <Servo.h>
          Servo servo;
#include <Wire.h>


// PINOUTS \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\


#define hallSensorPin 2 // Interrupt-pin f r hall-effect sensorn
#define atomizerPin 3   // F r vattenatomizern (HIGH = p )
#define solenoidPin 5   // F r solenoidventilstyrningen (HIGH = ventil  ppen )
#define motorStepPin 6  // F r att ta steg med stegmotorn (HIGH -> LOW = 1 steg)
#define motorDirPin 7   // Anv nds f r val av rotationsriktning p  stegmotorn
#define servoPin 8      // F r styrsignal  t  servon


// KONSTANTER \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#define stepsPerRevolution 1036 // Antal steg per varv med stegmotorn & v xell da
#define motorSpeedSlow 2000 // 4000 mikrosekunder mellan varje steg
#define motorSpeedFast 500 // 1000 mikrosekunder mellan varje steg
#define servoClosed 115 // Vinkeln d r servot h ller luckan st ngd
#define servoOpen 34 // Vinkeln d r servot h ller luckan  ppen

```

```

// VARIABLER \\\\\\\\\\\\\\\\\\\\\\\


float r = 20000; // Detta styr systemets genomsnittliga CO2-halt (i ppm) // <----

volatile bool hallSensorDetect = false;

bool isDataReady = false;
bool firstRun = true;
bool ventilate = false;

float co2InjectionDuration;
float temperature = 0.0f;
float humidity = 0.0f;

unsigned long timeKeeper;

int derivataHistory[10] = {0,0,0,0,0,0,0,0,0,0};
int n = 0;

uint16_t error;
uint16_t serial0;
uint16_t serial1;
uint16_t serial2;
uint16_t CO2 = 0;
uint16_t prevCO2 = 0;

char errorMessage[256];


// FUNKTIONER \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

void printUint16Hex(uint16_t value) {
    Serial.print(value < 4096 ? "0" : "");
    Serial.print(value < 256 ? "0" : "");
    Serial.print(value < 16 ? "0" : "");
    Serial.print(value, HEX);
}

void printSerialNumber(uint16_t serial0, uint16_t serial1, uint16_t serial2) {
    Serial.print("Serial:␣0x");
    printUint16Hex(serial0);
```

```
    printUint16Hex(serial1);
    printUint16Hex(serial2);
    Serial.println();
}

void hallSensor() {
    if (digitalRead(hallSensorPin) == HIGH) {
        hallSensorDetect = true;
    } else {
        hallSensorDetect = false;
    }
}

void motorStep(int motorStepDelay) {
    digitalWrite(motorStepPin, HIGH);
    delayMicroseconds(motorStepDelay);
    digitalWrite(motorStepPin, LOW);
    delayMicroseconds(motorStepDelay);
}

void co2InjectionRegulator(uint16_t CO2, unsigned long now) {
    const float m = 2004; // Ber knat genom tester
    const float k = 359; // Ber knat genom tester

    if (firstRun == true) {
        co2InjectionDuration = (r - m)/k;

        if (co2InjectionDuration < 0) {
            co2InjectionDuration = 2;
        }

        firstRun = false;
    }

    // Delta tid
    unsigned long delta_time = now - timeKeeper;
    timeKeeper = now;

    // Delta CO2
    int16_t delta_CO2 = CO2 - prevCO2;
    prevCO2 = CO2;

    int newVal = delta_CO2;

    if (n <= 9) {
```

```
        derivataHistory[n] = newVal;
        n++;
    } else {
        for (int i = 0; i < 9; i++) {
            derivataHistory[i] = derivataHistory[i+1];
        }

        derivataHistory[9] = newVal;
    }

    if (n == 10) {
        int s = 0;
        for (int i = 0; i < 10; i++) {
            s += derivataHistory[i];
        }

        if (abs(s) < 500) {
            int e = r - CO2;

            double e_rel = e / (double)r;

            float K = 1;

            if (abs(e) > 2000) {
                //K = 1.5;
            } else if (abs(e) < 500) {
                K = 0.5;
            }

            co2InjectionDuration += K*e_rel*co2InjectionDuration;

            if (co2InjectionDuration < 0) {
                co2InjectionDuration = 2;
            }

            for (int i = 0; i < 10; i++) {
                derivataHistory[i] = 0;
            }
            n = 0;
        }
    }
}

void setup() {

```

```
//  SETUP: Denna funktion k rs vid start av Arduino-kortet och sedan aldrig mer.

    pinMode(atomizerPin, OUTPUT);
    pinMode(solenoidPin, OUTPUT);
    pinMode(motorStepPin, OUTPUT);
    pinMode(motorDirPin, OUTPUT);

    servo.attach(servoPin);
    servo.write(servoClosed);

    attachInterrupt(digitalPinToInterrupt(hallSensorPin), hallSensor, CHANGE);

    Serial.begin(115200);
    while (!Serial) {
        delay(100);
    }

    Serial.print("CO2-m l:␣");
    Serial.print(r);
    Serial.println("␣ppm");
    Serial.println("|␣Uppm tt␣CO2␣|␣Pulsl ngd␣|␣delta␣tid␣|␣total␣tid␣|");
    Serial.println("␣-------------------------------------------------␣");

    Wire.begin();
    scd4x.begin(Wire);

    // stop potentially previously started measurement
    error = scd4x.stopPeriodicMeasurement();
    if (error) {
        Serial.print("Error␣trying␣to␣execute␣stopPeriodicMeasurement():␣");
        errorToString(error, errorMessage, 256);
        Serial.println(errorMessage);
    }

    error = scd4x.getSerialNumber(serial0, serial1, serial2);
    if (error) {
        Serial.print("Error␣trying␣to␣execute␣getSerialNumber():␣");
        errorToString(error, errorMessage, 256);
```

6

```
        Serial.println(errorMessage);
    } else {
        printSerialNumber(serial0, serial1, serial2);
    }

    // Start Measurement
    error = scd4x.startPeriodicMeasurement();
    if (error) {
        Serial.print("Error␣trying␣to␣execute␣startPeriodicMeasurement():␣");
        errorToString(error, errorMessage, 256);
        Serial.println(errorMessage);
    }

    Serial.println("Waiting␣for␣first␣measurement...␣(5␣sec)");

    digitalWrite(motorDirPin, LOW); // K r motorn motsols

    if (hallSensorDetect == true) {
        for (int i = 0; i < (stepsPerRevolution/2); i++) {
            motorStep(motorSpeedSlow);
        }
    }

    if (ventilate == true) {
        servo.write(servoOpen);
        for (int i = 0; i < (30*stepsPerRevolution); i++) {
            motorStep(motorSpeedFast);
        }
        servo.write(servoClosed);
    }

    while (hallSensorDetect == false){
        motorStep(motorSpeedSlow);
    }

    digitalWrite(atomizerPin, HIGH); // S l  p atomizern
}

void loop() {
    // Read Measurement
```

```
    error = scd4x.getDataReadyFlag(isDataReady);
    if (error) {
        Serial.print("Error␣trying␣to␣execute␣
            readMeasurement():␣");
        errorToString(error, errorMessage, 256);
        Serial.println(errorMessage);
        return;
    }

    if (!isDataReady) {
        return;
    }

    error = scd4x.readMeasurement(CO2, temperature,
        humidity);
    if (error) {
        Serial.print("Error␣trying␣to␣execute␣
            readMeasurement():␣");
        errorToString(error, errorMessage, 256);
        Serial.println(errorMessage);
    } else if (CO2 == 0) {
        Serial.println("Invalid␣sample␣detected,␣skipping
            .");
    }

    servo.write(servoOpen);

    delay(100);

    // Kolla om motorn  r  p   r tt plats. Om inte,
        r tta till.
    while (hallSensorDetect == false) {
        motorStep(motorSpeedSlow);
    }

    // K r motorn snabbt halvt varv
    for (int i = 0; i < stepsPerRevolution/2;  i++) {
        motorStep(motorSpeedFast);
    }

    delay(1000);

    for (int i = 0; i < stepsPerRevolution/2;  i++) {
        motorStep(motorSpeedFast);
    }

```

```
    servo.write(servoClosed);

    co2InjectionRegulator(CO2, millis());

    delay(500); // Kort paus innan CO2 sprutas in, ger servot en chans att st nga luckan.

    if (co2InjectionDuration > 2 && co2InjectionDuration < 100 && r != 0) {
        digitalWrite(solenoidPin,HIGH);
        delay(co2InjectionDuration);
        digitalWrite(solenoidPin,LOW);
    }
}
```

TRITA-ITM-EX- 2022:87

www.kth.se