3D-Workbench

Design and Development of a 3-Dimension Computer Numerical Controlled Machine

HAMK
UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Degree Programme in Automation Engineering

Valkeakoski, Spring 2015

Andrei Mircea Sandru

HAMK
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Valkeakoski
Degree Programme in Automation Engineering

Author Andrei Sandru **Year** 2015

Subject of Bachelor's thesis Design and Development of a 3-Dimension Computer Numerical Controlled Machine

ABSTRACT

The purpose of this thesis was to examine and develop a multipurpose Computer Numerical Controlled (CNC) device which would satisfy industrial requirements, but could also be implemented at universities for students to improve and apply their knowledge in different scopes. The topic was specifically chosen because of its close relation to a summer job at a metal factory the author completed and his personal fascination with 3D printers.

The project presented in this thesis was commissioned by HAMK University of Applied Sciences. The design and development of the prototype took place in the automation laboratory of HAMK UAS, Valkeakoski unit. Literature and product documentation established the main sources of information, although online resources were used as well. At the first stage, a research was carried out concerning 3D-printing related topics, such as interpolation and G-Code. Afterwards, a suitable control and motion system needed to be found. Once a list with suitable components was established, the machine design could take place. By using the design tool Autodesk Inventor it was possible to obtain a 3D model of the device.

Following the design, a prototype was built. A number of challenges were faced as major design changes had to be performed to the prototype. Still, the resulting prototype offered essentially the same functionality as the original design. For this prototype, a metal engraving tool was used at first for testing purposes, followed by a milling or drilling tool.

All in all, the results met and even surpassed the author's expectations. Recommended further improvements include an automated tool exchange system, additional tools, a reinforcement of the structure and the implementation of a user-friendly Human Machine Interface.

Keywords Design, 3D printing, CNC, servo motor, TwinCAT

Pages 58 p. + appendices 15 p.

LIST OF ABBREVIATIONS

CNC: Computer Numerical Control

3D: 3-Dimensions

CAD: Computer Aided Design

ISO: International Organization for Standardization

NC: Numerical Control

PTP: Point To Point

DIN: Deutsches Institut für Normung (German Institute for Standardization)

PLC: Programmable Logic Controller

PC: Personal Computer

CPU: Central Processing Unit

AC: Alternating Current

DC: Direct Current

OCT: One Cable Technology

ADS: Automation Device Specification

HMI: Human Machine Interface

CONTENTS

1 INTRODUCTION

The purpose of this thesis was to examine and develop a multipurpose device which could satisfy industrial requirements, but could also be implemented at universities for students to improve and apply their knowledge in different scopes. The topic was specifically chosen because of its close relation with a summer job at a metal factory the author completed. At this factory large, two-dimensional Computer Numerical Controlled (CNC) machines were used to cut out shapes from metal sheets.

The idea of a multipurpose 3D working device arose by combining an industrial CNC machine with a common personal 3D printer. The advantages and disadvantages of each type were taken into account from the first draft at the design stage, in order to maximize the performance of the machine while still being able to offer a cost-efficient product. The main advantages and disadvantages of industrial CNC machines and personal 3D printers are listed below. It is important to note that these two types of machines are not being compared together, but rather have their main benefits and weaknesses analysed.

CNC machines can work continuously, 24 hours a day, 365 days a year. The only moment when they need to be switched off is during regular maintenance. The parts are only designed once using any type of Computer Aided Design (CAD) program the operator is familiar with, and then produced as many times as required with a high accuracy and repeatability (i.e. same output specifications from the first to the last part produced). Because of the high accuracy and repeatability, parts produced with CNC techniques offer a very high degree of quality. Furthermore, CNC machines can work at a higher speed compared to manual labour, which results in a significant reduction of the production time for each part. The drawbacks of these systems include high initial costs and significantly high maintenance and service costs.

One of the greatest benefits of 3D personal or “home” printers is that they allow for rapid prototyping of nearly any shape at a reduced cost. In addition, manufacturers normally use low cost materials and parts resulting in affordable 3D printers. In contrast, the major detriments of 3D printers concern the limited material types which can be used for prototyping, improvable accuracy and the threat of printing copyrighted or dangerous items, such as weapons or knives.

On this thesis, the following aspects are covered:

- Basic theory and some mechanical concepts used.
- Description and selection of the components used.
- Three dimensional modelling.
- Software development and implementation.
- Building a prototype.

During the research project, a quantitative approach and research method was used. To answer the questions that arose from each step of the process, at first a theoretical study was carried out. Later at the prototyping

stage, the theory was put into practice and further developed or amended according to the empirical studies.

2 THEORETICAL BACKGROUND

2.1 Accuracy and repeatability

The International Organization for Standardization defines accuracy as the closeness of agreement between a test result and the accepted reference value (ISO 5725-1:1994). In other words, how closely a system can reach a commanded position.

The accuracy of the system presented in this thesis was determined by the control resolution offered by the encoders attached to the motors (spatial resolution) and the mechanical construction of the system (distributed mechanical inaccuracies). Both elements are further developed in their own chapters. Figure 1 presents the idea of accuracy in a system: *target point* stands for the commanded point; *spatial resolution* stands for the minimum, controlled step the system can perform; *accuracy* stands for the distance between the closest step and the *target point*; and *distribution of mechanical inaccuracies* indicates the random inaccuracies that can occur because of mechanical factors.

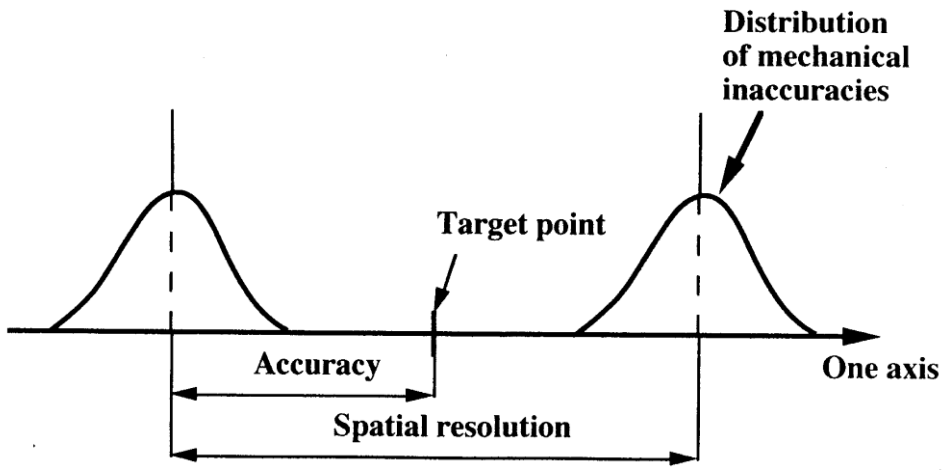


Figure 1 Accuracy of a system in one axis (Wahjudi 1999).

Another key characteristic of the system is its repeatability. Repeatability refers in this case to how well the system performs when commanded to return to a programmed position. As seen in Figure 2, repeatability forms a curve indicating the probable *return position* of the system when commanded to move to a *programmed point*. The difference between the *return position* and the *programmed point* indicates the *repeatability error*.

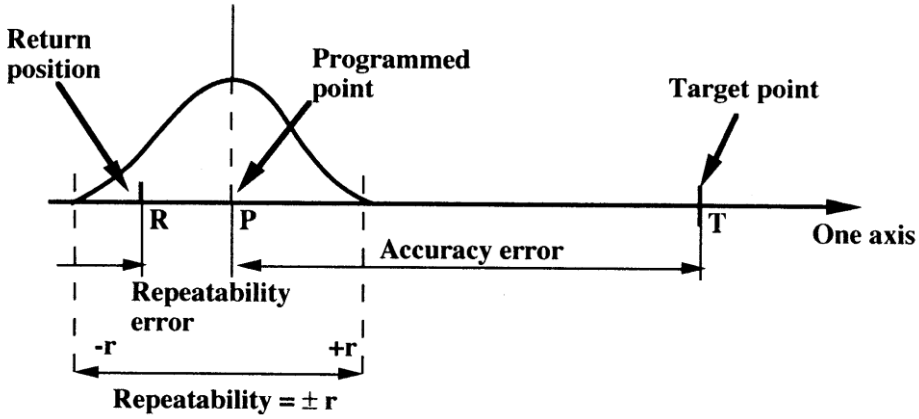


Figure 2 Repeatability of a system in one axis (Wahjudi 1999).

Accuracy and repeatability are important concepts, since they may determine not only the sales success of the system, but also the profit margin that can be obtained from each machine sold and the customer's satisfaction.

2.2 Interpolation

Cleve B. Moler (2004) describes interpolation as the process of defining a function that takes on specified values at specified points.

In other words, interpolation refers to the procedure of obtaining new data points from within a range of discrete known data points, creating a path between these points. Therefore, a three dimensional interpolation refers to the method of obtaining new values for the X, Y and Z axes from within a set of known points. Two main methods highlight among others: linear interpolation and polynomial interpolation.

It is common knowledge that any two points determine a straight line between them. From a mathematical point of view, two points with given coordinates in space determine a formula whose graphical representation is a straight line passing through the given points. This method is known as linear interpolation. When applying this method to a data set, a continuous line passing through each point in the data set can be obtained as shown in Figure 3.

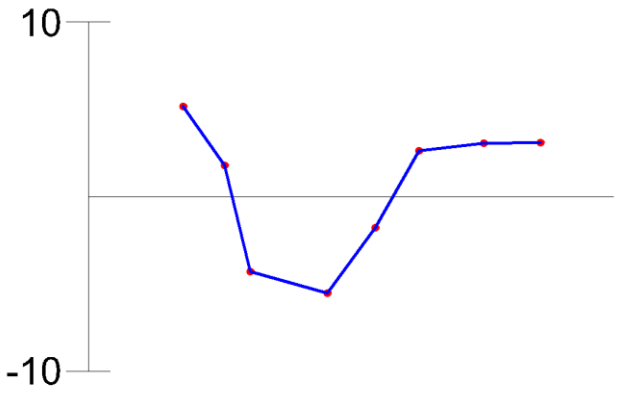


Figure 3 Example of concatenated linear interpolation.

Polynomial interpolation on the other hand is the search for a polynomial of *n* degree which goes through each point of a given data set. Figure 4 presents the idea behind polynomial interpolation: given a data set (red dots), a polynomial must be found such that its graphical representation (blue line) goes through each of the points in the data set.

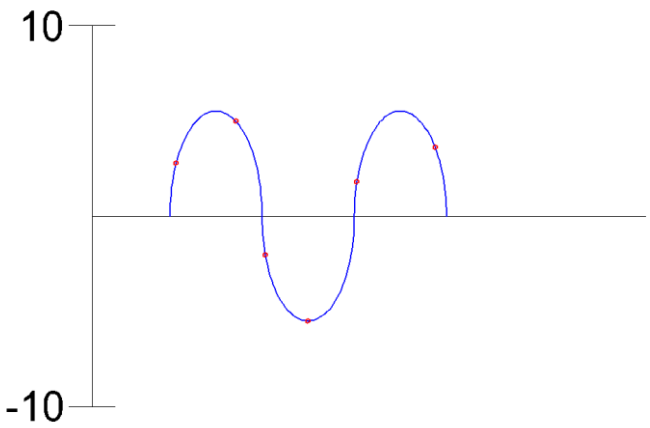


Figure 4 Example of polynomial interpolation.

Why is this important for the present project? In the case of 3D printing, a three dimensional model of the virtual object is the starting point. The surface of this model is in fact made up of small triangles; the smaller the triangles the smoother the surface, as illustrated in Figure 5 (Bourke 1992). Next, an algorithm is used to detect the intersection between the horizontal "slicing" plane and the triangles' vertices, obtaining different layers made of points. The algorithm used for interpolating between these points together with the number of points determines the accuracy (i.e. how close the end result is to the required product). As an example, the slicing process used for common 3D printers is shown in Figure 6.

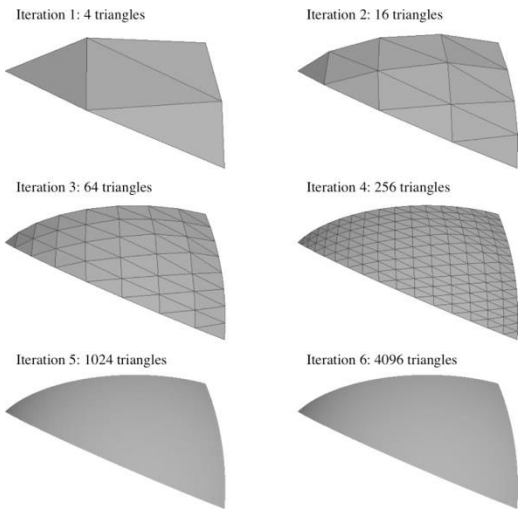


Figure 5 Comparison between the number of triangles used in the surface of a model and its smoothness (Bourke 1992).

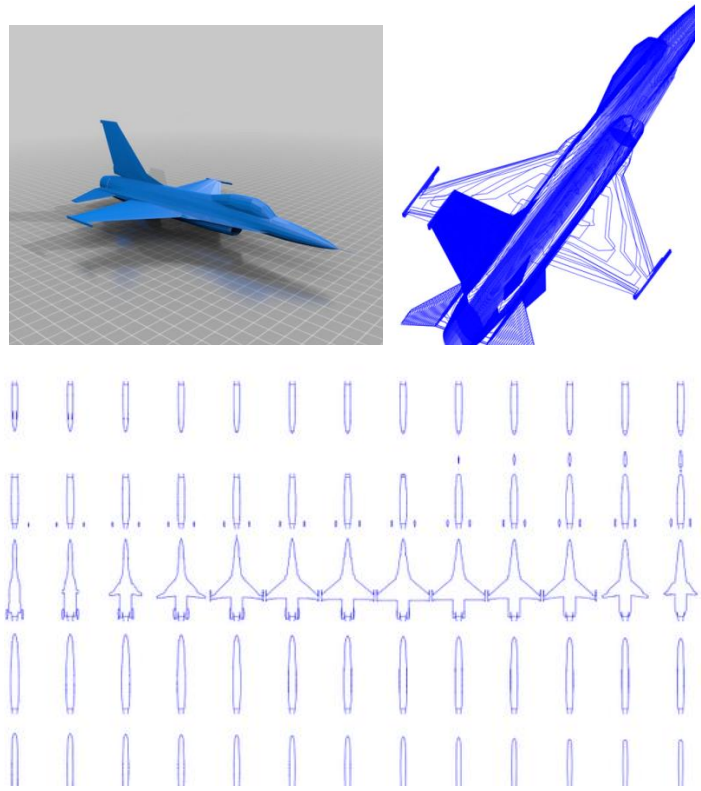


Figure 6 Slicing process of a 3D model (Gonen 2013).

In TwinCAT, the programming and working environment selected for this thesis project, the interpolation is carried out by the NC module, which is a Numerical Controlled system designed for interpolated path movements and integrated in the new TwinCAT 3 as an extension to TC3 PLC/NC PTP (Point To Point). It opens the possibility to perform movements with up to 3 interpolated path axes and geometry functions in 3D space. (Beckhoff 2014a.)

2.3 G-Code language

G-Code refers to a programming language widely used in industry for controlling automated machining equipment, such as CNC machines. At the present time, it is also the primary programming language for 3D printers. In essence, G-Code is formed by a set of instructions which tell the machine what to do. A short example of G-Code is presented next:

```
% Example program
% Defining parameters
N0 R0=3000
N0 R1=2400
% XY plane selection
N0 G17
% Absolute programming
N0 G90
% Preparatory command
N10 G1 X=60.0 Y=30.0 F=R1
% Miscellaneous commands
N10 M50
N20 G1 X=94.93 Y=48.36 F=R1
N20 M51
N20 G2 I=2.51 J=2.14 F=R0
% End
M30
```

In TwinCAT, the NC Interpreter accepts G-Code with a syntax that follows the guidelines established in DIN 66025, with an additional extension that includes some useful functions, such as: techniques for subroutines and jumps, programmed loops, zero offset shifts, tool compensations, tools and M and H functions. Working in three dimensions, the interpreter supports the following geometries: straight lines in space, circles in all main planes, circles in space, helices with base circles in the main planes and Bezier splines. (Beckhoff 2014a.)

2.4 Programmable Logic Controllers

2.4.1 Controllers

Before fully entering the vast world of Programmable Logic Controllers (referred simply as “PLC” or “PLCs” in the future), a proper definition and understanding of what a controller is and what it does is required.

An automated control system is used when it is preferred or required that a system performs certain actions without user interaction; bearing in mind reasons of security, commodity, efficiency, speed, etc. Examples illustrating this can be found in all kind of environments: automatic disconnection of an overheated grinder, preventing it to continue operating unless its motor temperature decreases below certain value; escalators working only when a person approaches them, thus reducing mechanical wear and energy costs; a resistance spot welding machine that automatically welds two parts together when the operator places them in the correct place. Figure 7 depicts a graphical design of the technique mentioned in the last example, without including any automation.

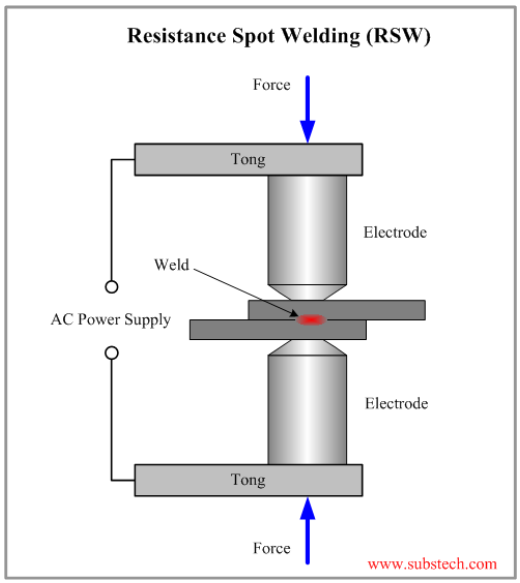


Figure 7 Resistance Spot Welding system (Dr. D. Kopeliovich n.d.).

When the operator places the two pieces in the correct place, a sensor is activated and the upper part of the welding machine descends until applying a predetermined pressure force on the welding spot, activating a second sensor. As soon as the second sensor is active, the welding machine stops descending, activates the welding current arc and a timer (usually of few second). When the timer reaches zero, the welding arc is stopped and the welding machine ascends to the original position. This cycle could be achieved by means of electrical circuits and wirings. As long as the process cycle remains the same, this solution appears to be ideal. Reliable, simple, cost efficient. But, what if the needs of the welding process change? What if, it is desired to fully automate the process using a robotic arm? Using the traditional electrical and wiring system would require a full update of the whole electric and wiring system in the first case, and as for the second case, the cost would render the sole idea unpractical and uneconomical.

Instead, using a programmable microchip or microcontroller to operate the whole system means a simple change in the program of the controller can modify the whole behaviour of the machine, thus reducing costs and increasing the flexibility of the system. In addition, increasing complexity tasks can be implemented. Continuing with the resistance spot welding machine, a detailed example can be found in most of nowadays car factories, where these types of welding systems are attached to robotic arms. These systems can weld the complete chassis of a car in a much shorter time period than any other human could. In case there are changes in the car model, downloading an updated version of the program to the robotic arm's memory can adapt the system to perform as required for the new chassis model.

2.4.2 Programmable Logic Controllers

William Bolton (2009, 3) offers a complete definition and description of what a PLC is: it refers to a special type of microprocessor based controller that includes all the components a microcontroller has and can perform functions such as logic, arithmetic, sequencing, timing and counting, in order to control machines and processes. These functions are pre-programmed "orders" in the controller's non-volatile memory (memory which holds its data even without power supply). Figure 8 presents a general idea of PLC. Inputs and outputs refer to digital I/O and in addition to every type of connection between the PLC and another system.

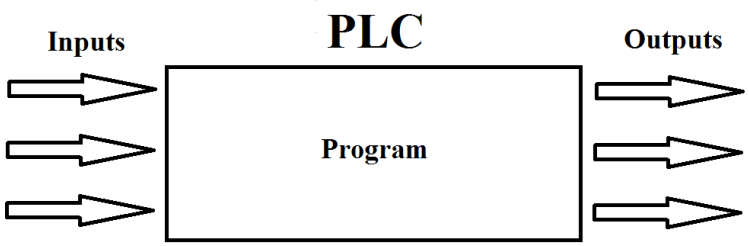


Figure 8 General idea of a programmable logic controller.

In order to program a PLC, a general approach would be: first, the operator needs to establish a connection between the PLC and a personal computer by means of a suitable connection type and cable; second, he or she will design the code containing the required instructions for the PLC to perform the necessary functions; third, if there are no errors in the compiling of the program the operator may proceed to upload the program to the PLC's memory and proceed with the test runs. First and second steps are interchangeable in order.

Consequently, if it looks and can perform similar to a generic microprocessor-based controller, what makes a PLC so special? A PLC is designed in a way that engineers from non-computer science degrees can program and operate it. In other words, PLC producers assume a limited knowledge of computers and computer programming languages from the installers. To achieve this, PLC manufacturers include software for programming the PLCs with a rather simple, intuitive interface and language. There are different types of programming languages for PLCs, but nowadays they tend to be more and more standardized and even allow different programming languages to be used at once. This provides PLCs with their most powerful argument compared to traditional wiring systems: they can be programmed over and over again, easily adapting them to new tasks or working conditions. (Bolton 2009, 3.)

Comparing a PLC and a Personal Computer's operation, one similarity that can be observed at first is their way of handling tasks: they both have a cycle running internally taking care of the tasks in sequence, one after another. Even though it may seem that all the actions are performed at once, this is a result of the incredibly short cycle time inside the PLC's microprocessor. In fact, nowadays' latest PLC models can implement powerful microprocessors and run simplified versions of Windows OS,

called Windows CE or even a full version of Windows 7 (Beckhoff 2014a). Some manufacturers' programming tools can be used to simulate a PLC inside a Personal Computer (PC), reducing costs where possible, although this is an occasionally acceptable option, as can be seen next.

If PLCs are a simplified version of PCs, in what aspects do they differ? First of all, PLCs are designed to withstand harsh industrial conditions: dust, vibrations, temperature, humidity and noise; conditions where a typical personal computer would simply stop working after some time. Another difference can be deduced from their purpose: a PC is optimized for calculus and display tasks, while PLCs are optimized for control tasks and therefore include interfacing for inputs and outputs. One last difference was already presented before and regards the low skills level required to program a PLC, compared to highly demanded skills to program a PC.

2.4.3 Hardware of a PLC

As mentioned before, basic PLCs have similar components with microcontrollers. More advanced PLCs include components traditionally more related to personal computers, such as powerful processors, extended memory and even hard drives (Beckhoff 2014a).

In general terms, PLCs include the following components:

- Processing unit (CPU): the "brains" of the PLC. It interprets the inputs and takes decisions regarding the outputs, based on the program stored on non-volatile memory, this is, the memory that doesn't get erased once the power supply is interrupted. The program is usually stored on ROM memories, but can also be programmed on hard drives or memory cards. The communication within the PLC is attained through buses. A bus is simply a physical path of connection between two components, for example between the CPU and memory modules, or between the CPU and inputs and outputs terminals.
- Power supply: supplies the required power for the PLC to operate. Usually PLCs use 24 volts logic to communicate with other devices and systems, therefore a second power adapter located within the PLC is required to power the CPU, which normally uses 5 volts logic or even 3.3 volts.
- Input/Output units: allows the communication between the PLC and external devices and systems. Each input and output point has its own address for the CPU to control. The communication however is hardly direct: the signal is conditioned and adapted to the voltage levels required by the CPU. Electrical isolation is usually achieved by means of optocouplers, which consist of a light emitting diode separated with a gap from a photo-sensor. When a signal activates the diode, the sensor detects this change and acts similar to a closed switch, allowing the signal to continue but at the same time separating both circuits. Optocouplers allow a wide range of input voltages, conditioning them to the same level. In order to accommodate higher power demanding outputs, such as to control a small DC motor, extra components are required. Regarding the components used to control the output, outputs can be of relay type, transistor type or triac type.

2.4.4 PLC Systems

In essence there are three basic types of PLCs, defined by their physical design: a single box, modular/rack or PC based controllers.

The single box or brick type includes all the necessary components for a small system. Processor, memory, power supply and input/output units are enclosed in the same package. They usually have a limited number of I/O connection points and enough memory for a few hundreds instructions. In case more inputs and/or outputs are required, a special bus is implemented to connect with other units, such as bus couplers.

The modular systems separate different components in units or modules designed to fit in racks, therefore they include different modules for power supply, CPU and Input/Output units. The first advantage of this type of systems is their flexibility. The person in charge of designing the system can decide how many I/O cards are needed and plug or connect only those. One example of this type of systems can be found in Beckhoff's catalogue. Their system is based on modular cards, which can easily be connected or replaced according to needs.

The last type refers to personal computers used as PLCs. This is achieved by simulating a virtual PLC runtime inside a personal computer. Personal computers have none of the required input and output connection points as PLC do, therefore there is a need for an external device to interface between a PC and other systems. This can be done by means of a bus coupler. Bus couplers interconnect a controller (PLC or PC) with I/O terminals; activating the required outputs, reading inputs, sending and receiving data. Consequently, bus couplers merely follow orders; they do not have the sufficient processing power to make decisions.

2.5 Servomotors and drives

In this chapter, a brief description is presented of how most common motors are built and work, with a particular detailed explanation of synchronous servomotors contained within.

A generally accepted definition stands that an electric motor is a device which converts electrical energy into mechanical energy. Most electric motors operate through interaction between magnetic fields and winding currents to produce mechanical forces, although electrostatic motors use electrostatic forces (force of attraction or repulsion between electric charges). A reverse operating device converting mechanical energy into electrical energy is also possible, and is called a generator. Numerous types of electrical motors can be run in generating or braking modes and vice versa.

Figure 9 Various electric motors compared to a 9 V battery (Seward & Zeigler 2012).

Nowadays applications for electrical motors are countless. From the tiny motors inside wrist watches, to the immense motors powering modern industry; they are all based in the physical principle of production of mechanical force through electromagnetism discovered in 1821. Figure 9 presents some commonly found motor types. Other type of devices may produce mechanical forces in some way, but are not included as electrical motors. This is the case for example of speakers and solenoids. The principle is similar, but the end result is totally different: they do not generate a rotational force.

Moreover, science has made possible through the development of new materials or techniques the possibility to increase even further the already vast scale of electrical motors. One example of how far science has travelled resides within the walls of Tufts' University School of Arts and Sciences. Chemists from this university have managed to develop and test world's smallest electric motor, made from a single molecule. This motor is merely 1 nanometre across, while a human hair is 60000 nanometres wide. (Tierney, Murphy, Jewell, Baber, Iski, Khodaverdian, McGuire, Klebanov & Sykes 2011.)

Table 1 summarizes some of the most common electric motor types. There are many ways for electric motors to be classified: by their source of electric power, internal construction, application, type of motion they give. A traditional way of division has been between AC and DC motors.

Table 1 Comparison of motor types (Seward & Zeigler 2012).

Type	Advantages	Disadvantages	Typical Application	Typical Drive
AC poly-phase induction squirrel-cage	Low cost, long life, high efficiency, large ratings available (to 1 MW or more), large number of standardized types	Starting inrush current can be high, speed control requires variable frequency source	Pumps, fans, blowers, conveyors, compressors	Poly-phase AC, variable frequency AC
Shaded-pole motor	Low cost Long life	Rotation slips from frequency Low starting torque Small ratings low efficiency	Fans, appliances, record players	Single phase AC
AC Induction (split-phase capacitor)	High power high starting torque	Rotation slips from frequency Starting switch Required	Appliances Stationary Power Tools	Single phase AC
Universal motor	High starting torque, compact, high speed	Maintenance (brushes) lifespan Only small ratings Economic	Drill, blender, vacuum cleaner, insulation blowers	Single phase AC or DC
Single phase AC or DC	Rotation in-sync with freq - hence no slip	More expensive	Industrial motors Clocks Audio turntables tape drives	Poly-phase AC
Stepper DC	Precision positioning High holding Torque	High initial cost Requires a controller	Positioning in printers and floppy drives	DC
Brushless DC	Long lifespan, low maintenance High efficiency	High initial cost Requires a controller	Hard drives CD/DVD players electric vehicles	DC
Brushed DC	Simple speed Control	Maintenance (brushes) Medium lifespan Costly commutator and brushes	Steel mills Paper making machines Treadmill exercisers automotive accessories	Direct DC or PWM
Pancake DC	Compact design Simple speed Control	Medium cost Medium lifespan	Office Equip Fans/Pumps	Direct DC or PWM

A more consistent way to divide motors takes into account the required synchronization between a moving magnetic field and a moving current sheet in order to produce an average torque. This leads to a distinction between asynchronous and synchronous types of motors. Asynchronous motor types require a slip between the moving magnetic field and the winding set to induce current in the winding set by mutual inductance, or in other

words, the input current creates a rotating magnetic field in the motor's stator, which at the same time induces currents in the rotor's conductive bars and results in a mutual attraction between the rotating magnetic field in the stator and the induced magnetic field in the rotor. Usually the common AC induction motors are referred as examples of asynchronous motor types. In contrast, synchronous motors do not require the slip, are all AC motor types and their rotor rotates at the same speed as the rotating magnetic field inside the stator. (Seward & Zeigler 2012.)

Permanent-magnet motors rely on permanent magnets to provide the magnetic field against which a rotating magnetic field interacts in order to produce torque. The permanent magnets can be located either in the stator or the rotor. The strength of the magnetic field produced by the permanent-magnet determines the size and electrical power needed for the motor to produce a determined speed-torque characteristic graph. Therefore, in order to reduce the size and weight or to improve the power of permanent-magnet motor, powerful magnets made of strategic materials such as neodymium are used.

Linear motors distinguish themselves from other types of electrical motors with their design. In a linear motor, the field winding is extended flat and therefore it produces a linear mechanical force. They can be further divided in high acceleration (used for example in Gauss gun) or low acceleration (used to power some of the most advanced trains).

2.5.1 DC motors

This type of motors runs on DC (direct current) electric sources. Common DC motors include brushed motors (internally commuted) and brushless motors (externally commuted).

Brushed DC motors rely on a split ring commutator with brushes in order to oscillate the current inside the wound rotor or armature, which interacts with a wound or permanent magnet stator to produce rotational torque. The commutator powers the wound rotor through the brushes, and causes the current to be switched as the rotor turns, not allowing the magnetic poles of the rotor to align with the magnetic poles of the stator. Most of the limitations arise from the need for brushes in the design. These brushes need to press against the commutator generating friction, sparks, RFI and even short circuiting coil ends. Therefore these brushes will eventually wear off and need to be replaced. In addition, the output speed of the motor needs to be limited, as excessive speed would cause the brushes to overheat, erode or even melt. A cross section of one of the most common DC motor which can be found in many children toys is represented in Figure 10.

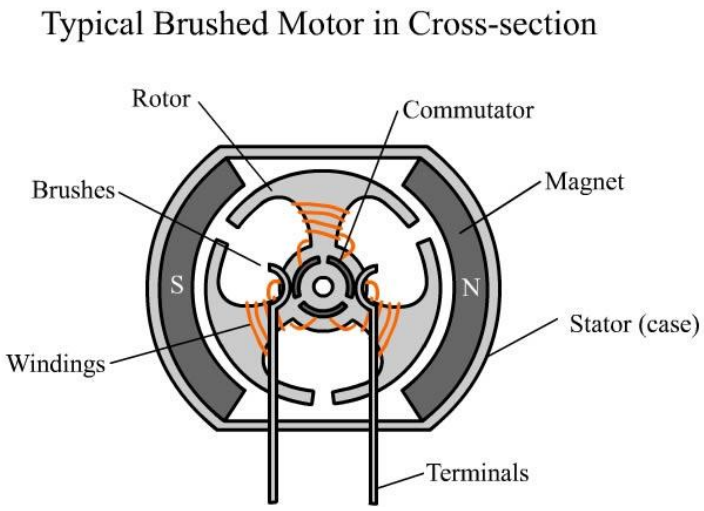


Figure 10 Common DC motor found in toys (Helms 2011).

In the brushless DC motors design, the rotation is achieved by means of an externally synchronized device to the rotor's position, which commutes the current inside the wound rotor. Brushless DC motors often use a permanent-magnet external rotor, three phases of driving coils, one or more Hall Effect sensors and the required drive electronics. The drive electronics sense the position of the rotor by means of the sensors and activate the required coil. Eliminating the commuter solves many of the problems brushed DC motors carry. For instance, brushless DC motors produce no sparks, require less maintenance and are quieter and cooler while running. One downside however is that they are more expensive. This type of drives is used in applications that require a precise speed control, such as computer hard drives. One of the latest applications for DC brushless motors involves powering the increasingly popular electric cars (Seward & Zeigler 2012).

Stepper motors, although resemble in design to three phase AC synchronous motors, use DC power. A stepper motor consists of a permanent-magnet rotor and a stationary field winding. The field winding usually consists of two sets of coils, which an external control circuit can directly control. The control circuit activates each set at a time, which leads the rotor to align its magnetic field to the coil's magnetic field. Therefore, the motor does not rotate continuously but it goes from one coil to another, it steps from one position to the next. If both coils are active, the rotor will position itself halfway between them. This control mode is called half-step. The amount of coils in the field or "steps" the motor has per revolution determines its control resolution (minimum angle it can turn and stop). It is also possible to achieve smooth rotation by controlling the amount of power in each coil. Given their design, stepper motors are able to rotate a specific angle (bearing in mind their control resolution) and therefore precisely control position, speed and acceleration. Stepper motors are commonly found in inkjet and laser printers, head positioning in DVD readers and more recently in consumer 3D printers.

Other forms of DC motors are coreless or ironless DC motors, which are constructed without any iron core and can produce great mechanical ac-

celerations; and the printed armature or pancake DC motors, which have the windings shaped as a disc running between high-density flux magnets arranged in a circle.

2.5.2 Universal motors

Universal motors refer to a specific type of motors designed to be able to operate on either AC or DC power sources. To achieve this, the field and armature windings are connected in series and therefore the current through them reverses synchronization, leading to alternating magnetic fields and therefore a mechanical force in one direction. Figure 11 offers a graphical description of the concept.

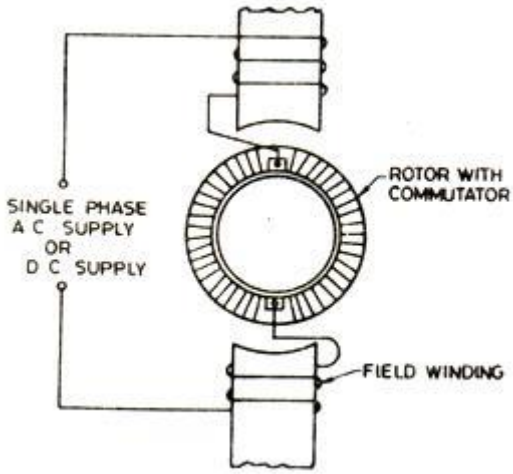


Figure 11 Universal motor's rotor and field windings connected in series (Universal motor n.d.)

One benefit of universal motors is that they can take advantage of some characteristics normally found in DC motors, such as high starting torque and compact design. In contrast to induction motors, which have their maximum speed determined by the power line's frequency, universal motors usually run at high speeds and may include electronic speed control, making them the ideal choice for home appliances such as vacuum cleaners, hair dryers, grinders, washing machines, etc. The drawbacks come from the need of a commutator with brushes, increasing maintenance demands and reducing life expectancy. Because of this, universal motors are commonly found in application which demand a high starting torque and intermediate use, as for example in blenders. In the case a universal motor runs with no significant load, there is a risk of the motor running at a higher speed than it was conceived, leading to mechanical damage. Another type of damage may arise in larger motors when a sudden load loss occurs. This can be avoided incorporating artificial loads to the motor such as a fan, which at the same time helps cooling down the armature and field windings. (Seward & Zeigler 2012).

2.5.3 AC motors

Two subcategories can be found in AC motors: induction or asynchronous and synchronous motors.

Induction motors relay on electromagnetic induction from the windings to the rotor in order to achieve mechanical power. In essence, induction motors resemble a rotating transformer considering the stator the primary side and the rotor the secondary side. A subdivision of induction motors further divides them into squirrel-cage and wound-rotor motors.

When used in an application where the load torque curve increases with speed, an induction motor will accelerate up to the speed where the torque produced by the motor equals that required by the load, thus increasing or decreasing the load will increase or decrease the motor's speed.

As mentioned before, synchronous motors do not rely on slip in order to produce mechanical force. Instead, their rotor's permanent magnet or field winding supplied with DC power generates a constant magnetic field, which spins in synchronization with the rotating magnetic field generated in the armature winding. In case the synchronization is lost because for example a great load, the motor will come to a standstill position (unable to spin, with the rotor locked).

Synchronous motors basically consist of the following parts, although larger ones may include additional components such as forced cooling systems or self-lubricating circuits for bearings: a stator (outer shell of the motor, carries the armature winding spatially distributed for poly-phase AC current, which generates the rotating magnetic field inside the motor), a rotor (rotating axis of the motor, carries a permanent magnet or the field winding supplied with DC current), slip rings (supply DC current in case the rotor has field windings) and the stator frame or enclosure (supports all the components).

Synchronous motors are divided into two major categories: non-excited and direct-current excited. With recent improvements in independent brushless excitation control of the rotor's winding set a third category can be included: brushless wound-rotor doubly-fed electric machines. This type of motors offers power factor correction, highest power density, highest potential torque density and low cost electronic controller among others. Non-excited synchronous motors can be further divided in permanent magnet, reluctance and hysteresis designs (last two employing self-starting circuits and therefore no external excitation supply). Direct-current excited motors' power rating start at 1hp and require a direct current supplied to their rotor windings using slip rings for excitation. (Seward & Zeigler 2012.)

Calculating the nominal speed of a synchronous motor is done following equation 1; where v is the speed of the rotor in Revolutions per Minute (rpm), f is the frequency in Hertz (Hz) of the power supply line and n represents the number of magnetic poles.

$$v = \frac{120*f}{n} \quad (1)$$

Because of the inertia of the rotor, synchronous motors cannot start by themselves. As soon as the motor is powered, the armature winding creates a magnetic field rotating with the line's frequency. The rotor however, because of inertia, cannot follow the instantaneous rotating speed of the armature magnetic field. To overcome this, different methods may be used: a separate motor ("pony motor") initially spins the rotor up to synchronization with the stator's rotating magnetic field; starting the motor as an induction type by shunting the windings or implementing induction motor like arrangements; or using a variable frequency drive and gradually increase the frequency up to the required speed. (Seward & Zeigler 2012.)

The advantages of synchronous motors over asynchronous ones are described below:

- If an adequate field current is applied, the load applied does not affect the motor's speed.
- Depending on their design, an accurate speed and position control can be implemented using open-loop control (for example, stepper motors).
- They hold their position while a DC current is being applied to the stator and rotor.
- They can achieve unity power factor and help improve the power factor of the whole installation (power factor of an AC circuit refers to the ratio between the real power used to do work and the apparent power supplied to the circuit).
- Used at low speeds, they can achieve increased electrical efficiency.
- Because of their design, they either run at the synchronous speed or they do not run at all (stall).

(Seward & Zeigler 2012.)

2.5.4 Position sensors

In order to monitor the rotor's position, sensors ought to be used. These sensors are called encoders. An encoder is a device that converts information from one format to another, and in this case, it converts mechanical position into electrical format suitable for a machine to understand. Some examples of encoders are: potentiometers, optical encoders and resolvers. The accuracy of a system is directly dependent of the encoder used and are related as follows: accuracy of a system is equal to half of the control resolution offered by the encoder.

Potentiometers offer a cheap method for positioning. They can be single-turn (less than 360ºdegrees allowed movement) or multi-turn. Some of the drawbacks of this type of systems become apparent as a result of their mechanical design. The movement of the wiper against the resistive material causes wear (and in time, lack of accuracy), the output is affected by temperature and humidity, and since the output of the potentiometer is of analog type, it requires an Analog to Digital Converter (ADC). Potentiometer

feedback systems can be found in inexpensive control systems, such as small servomotors used in Radio Control planes and boats. Because of their low reliability and high maintenance, potentiometers are not acceptable as positioning systems in medium and high end servomotor systems.

Optical encoders on the other hand do not require physical contact and therefore offer greater wear levels and higher reliability. They consist of one or more optical sensors paired together with a light emitter and a coded ring between them. Two types can be distinguished: incremental and absolute optical encoders.

Incremental optical encoders provide output pulses proportional to the rotation of the shaft and therefore cannot remember the position prior to power off. An external counter must be used to keep track of the number of pulses (and therefore the displacement) of the shaft. Consequently, the system must have a way to determine the initial position of the shaft and based on the information from the counter determine the actual position of the motor. This method is called homing and consists in driving the motor to a known position determined by, for example, optical sensors. Incremental optical encoders can be of two types: single channel or quadrature encoder. Single channel encoders use only one pair of light emitting diode and photo sensor and therefore they can only detect the displacement of the shaft, but not the direction. This type of encoders find applications in systems where there is only one way rotation, the rotation direction can be determined by other means or only an accurate measure of speed is required. Quadrature encoders on the other hand normally use two pairs of light emitting diodes and photo sensors, and output two signals (A and B) phased 90º apart. Based on the two signals, speed and direction can be determined. The maximum resolution of the encoder is determined by the number of “gaps” or divisions the rotor plate has. Some systems may include a third phase signal (Z), used as the origin signal. Figure 12 offers a graphical explanation of an incremental optical encoder. This type of encoders is used where precise control over speed and position is required.

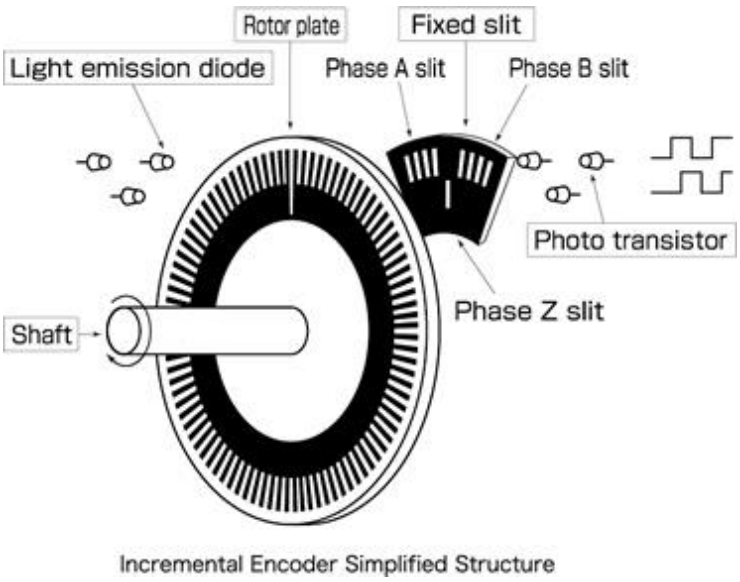


Figure 12 Working principle of incremental optical encoder (Tawagawa n.d.).

Absolute optical encoders usually have three or more pairs of sensors and emitters situated in a line perpendicular to the rotation axis. As the coded ring spins between the sensors, different patterns or binary codes emerge. The maximum resolution available is limited by the number of optical pairs of sensors and emitters. Figure 13 presents an example of the working principle of a simplified 8 bits absolute optical encoder. In this example, with 8 pairs of sensors and applying binary logic, 28 = 256 possible positions can be detected and therefore the system can detect movements of up to 1.40625º. One of the main advantages of absolute optical encoders is that it maintains the position information when power is removed and can immediately inform of the position at power up.

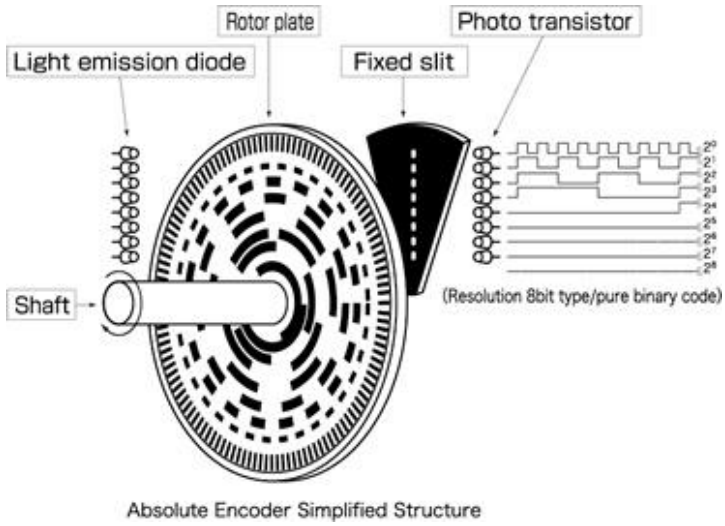


Figure 13 Working principle of absolute optical encoder (Tawagawa n.d.).

Resolvers transmit angular data electrically with a high degree of accuracy and are similar to variable transformers in which the coupling between windings varies with the rotor's position. They consist of two windings offset by 90º mounted around a stator. Resolvers are generally accepted as the most robust and long-term reliable in a wide range of operating environments among angular measurement devices. Figure 14 presents the internal construction of resolvers.

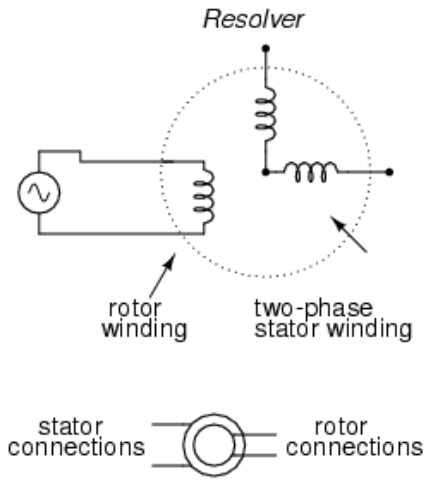


Figure 14 Electrical construction of a resolver (Small Electronic Thingies for All Kinds of Fun Stuff 2010).

2.5.5 Servo technology

In the past, servo drives were only used as auxiliary drives for secondary tasks, hence the name “servo” (from Latin “servus” which means slave). This was because of their inefficient analog control systems. In contrast, today’s rapid development of the industry of semiconductors and micro-controllers lead to servo drives with increased functionality and more and more applications where servo systems are used as the main drives. The three main types of servo drives (synchronous servomotors, asynchronous servomotors and synchronous linear motors) are currently used in the following industries: packaging technology, robotics, machine tools, handling systems, sheet metal processing, paper processing and material handling. (Sew-Eurodrive 2006, 7.)

A servomotor is used when effective control of position or speed is required. Basically, almost each type of motors can be used as a servomotor by attaching an encoder to their rotor and using a suitable control system, although depending on the requirements not all motors are suitable as servomotors for all types of applications. Servomotors differ from stepper motors in that the position and/or speed of a servo drive is constantly monitored and therefore the control system knows at any given moment the exact position of the motor. Even if external forces may have changed the rotor’s position, the control system can detect this change and drive the motor to the required position. Stepper motors rely on not missing steps and consequently, if for example an external force changes the actual position of the motor, the control system has no means to detect this change. To improve accuracy, home position switches may be used to calibrate the motor before each working sequence.

Motors planned to be used as servomotors must have their characteristics for speed, torque and power well documented. Whether the dynamic response is not important (slow servo loop), conventional AC or DC motors may be used together with a position or speed feedback device. When high dynamic response is expected from the system as in the case of a flying

saw or a Computer Numerical Control (CNC) machine, more specialized motor designs (e.g. coreless motors) are required to improve the overall performance of the closed-loop control. (Seward & Zeigler 2012.)

Specialized servomotors are motors which display high dynamics, high position accuracy and high overload capacity over a wide range of speeds; in addition to high speed accuracy, short acceleration and torque rise time, high static torque, small inertia, low weight and compact design. Common servomotors can be grouped as shown in Figure 15.

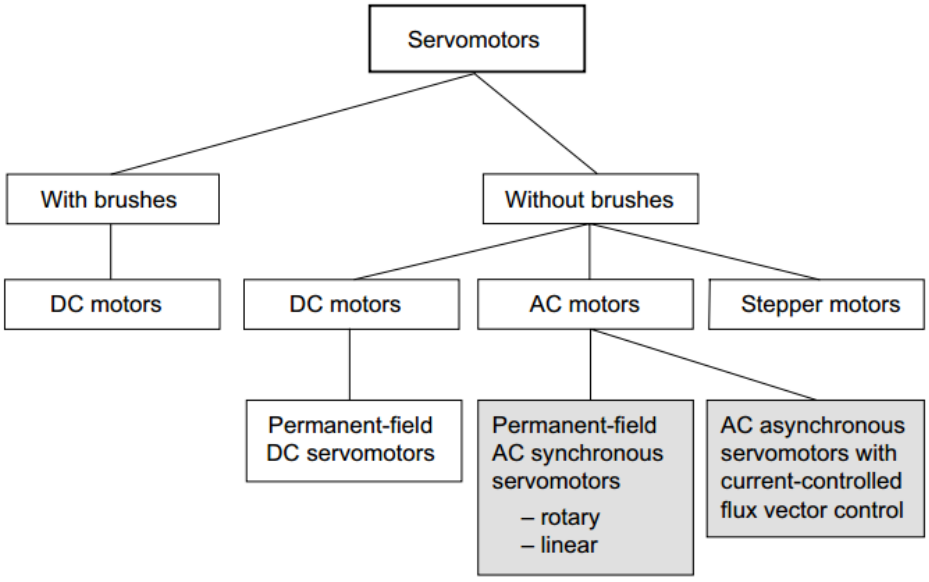


Figure 15 Overview of servomotors (Sew-Eurodrive 2006, 2).

Figure 16 displays a comparison between the features of synchronous and asynchronous servomotors. Because of their high power density, acceleration characteristics and ease for position control, synchronous permanent-magnet motors are the ideal choice for high demanding dynamic characteristics applications. As a result, this type of drives will be used for the present thesis project.

Features of synchronous servomotors	Features of asynchronous servomotors
High dynamics	Moderate to high dynamics
Moderately good control characteristics for large masses	Good control characteristics for large external masses
High overload capacity, up to 6 x	High overload capacity, up to 3 x
High thermal continuous load capacity throughout the entire speed range	High thermal continuous load capacity; depending on speed
Heat dissipation via convection, heat transmission and emission	Heat dissipation via fans
High speed quality	High speed quality
Static torque continuously available	Due to thermal load in the lower speed range that is too high, torque cannot be available continuously without a forced cooling fan
High speed setting range, 1:5000	High speed setting range, 1:5000
Torque ripple (cogging) at low speeds. See also the definition on page 89.	Almost no torque ripple (cogging). See also the definition on page 89.

Figure 16 Comparison between the features of synchronous and asynchronous servomotors (Sew-Eurodrive 2006, 12).

The basic design of most of nowadays synchronous servomotors consist of a rotor with permanent magnets, a stator with windings, a power connector or terminal box and an encoder connection. Some manufacturers offer models that combine the power and encoder connections in one single proprietary connector to reduce cable and installation costs, for example Beckhoff's One Cable Technology (OCT) (Beckhoff 2012a).

Connecting the servomotor to an appropriate servo inverter allows precise control of the rotating magnetic field inside the stator. In turn, this rotating field applies a magnetic force on the rotor's permanent magnet field, making it turn in synchronism. When a load is applied to the rotor, a "lag" (displacement angle) appears between the poles of the rotating magnetic field in the stator and the rotor.

Increasing the displacement angle increases the torque applied by the motor, reaching a peak torque at a displacement angle of 90º. Consequently, the stator pole must always lead by 90º while in motor operation and lag by 90º in regenerative operation for maximum torque to be obtained. If the displacement angle increases beyond 90º, the produced torque decreases and the motor enters an unstable working condition where it may remain stalled, causing thermal damage. Figure 17 offers a clear graphical observation of the current ratios inside the stator; [1] refers to the current space vector I (vectorial sum of the currents iu, iv, iw) and [2] represents the ratios in the stator with regard to the generation of torque at various points in time. (Sew-Eurodrive 2006, 18-19.)

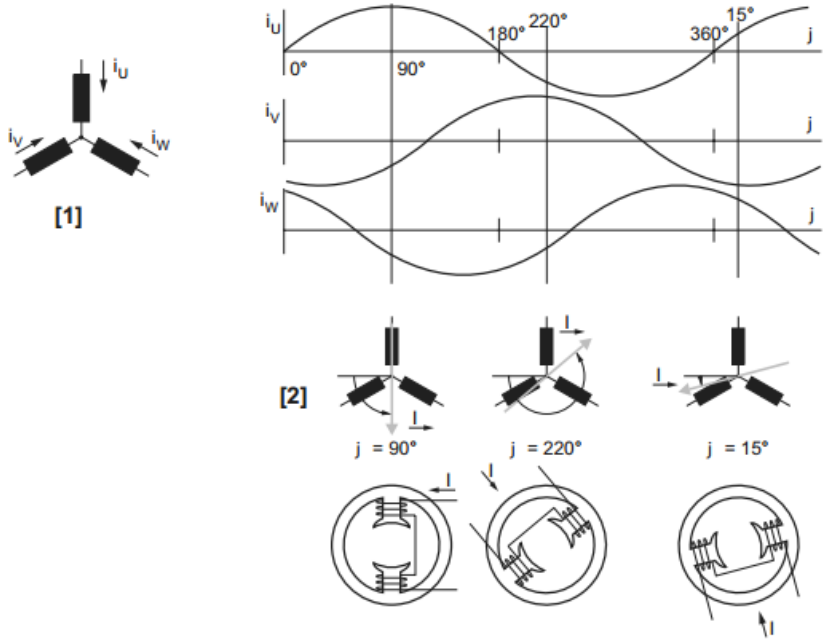


Figure 17 Current ratio in the stator (Sew-Eurodrive 2006, 19).

Additionally, some servomotors may include a factory pre-mounted electromechanical brake. Some of the applications for this feature include stopping loads, performing emergency stops, stopping machine units or holding position while power off.

2.5.6 Servo inverters

It was mentioned before that the speed of a synchronous motor is determined by the frequency of the power supply and the number of poles. Since the number of poles is determined by the motor's internal construction at factory, it is not possible to dynamically vary the speed by changing the number of poles. On the other hand, the rotor spins in synchronization with the rotating magnetic field inside the armature wielding, which is determined by the frequency of the power supply. Therefore, it is clear that by varying the frequency supplied to the motor, its speed can effectively be controlled. This is done using servo inverters.

Modern servo inverters take advantage of the developments in electronics area to provide powerful features, such as high control qualities, high dynamic properties, overload capacity, powerful microcontroller control with increasing PLC functions, complex functions (i.e. electronic cam, phase-synchronous operation, touch probe processing, torque control), flexible interfaces (i.e. analog and digital inputs and outputs, optional PCB slots for multiple encoder and fieldbus interfaces), increased range of supply voltages.

The working principle of the power section of a servo inverter is frequently based on the DC link amplifier. The DC link is directly generated in the converter section via a B6 diode from the three-phase supply line and then stored in capacitors (Figure 18). The total capacity of the capacitors in the

DC link determines the amount of energy it can accept. (Sew-Eurodrive 2006, 67.)

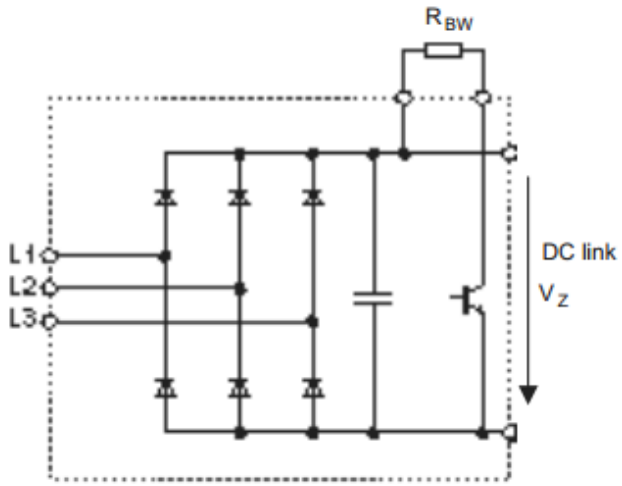


Figure 18 Circuit diagram of the DC link and B6 diode bridge converter (Sew-Eurodrive 2006, 67).

The DC link voltage is used to power the inverter (Figure 19). Using correct clocking and six IGBTs (Insulated-Gate Bipolar Transistors) a pulse-width modulated voltage is generated at the output of the drive and applied to the motor, generating the rotating magnetic field in the stator. Because of the motor and cable inductances, the current inside the stator is almost sinusoidal. Additionally, a diode is connected in inverse parallel to each IGBT preventing self-induced voltage from the motor to damage the inverter and at the same time conducing it back to the input of the inverter. (Sew-Eurodrive 2006, 68.)

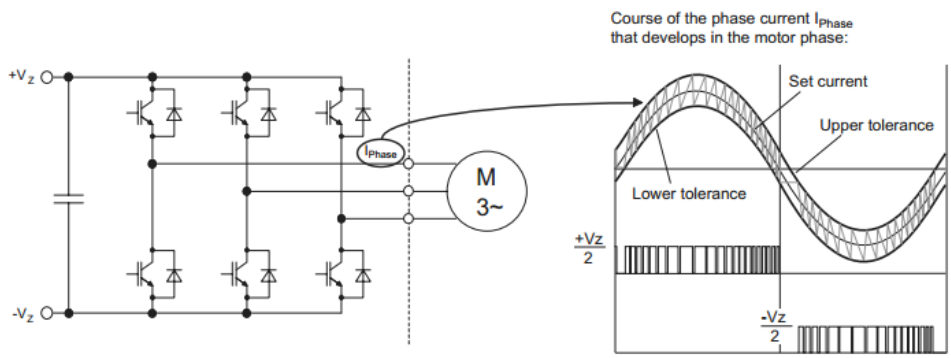


Figure 19 Block circuit diagram of the inverter (left) and pulse-width modulated voltage and current flow in motor (right) (Sew-Eurodrive 2006, 68).

2.6 TwinCAT

TwinCAT from Beckhoff is the programming and working software environment chosen for developing the project in this thesis.

Beckhoff developed a new global standard in 1986 when their PC-based control technology was released. Nowadays, with eXtended Automation Technology (XAT) Beckhoff has entered a new era, further developing the

previous version of TwinCAT and increasing the integration and interoperability among systems and programming languages. (Beckhoff 2012b.)

Figure 20 presents the main features included in TwinCAT 3.

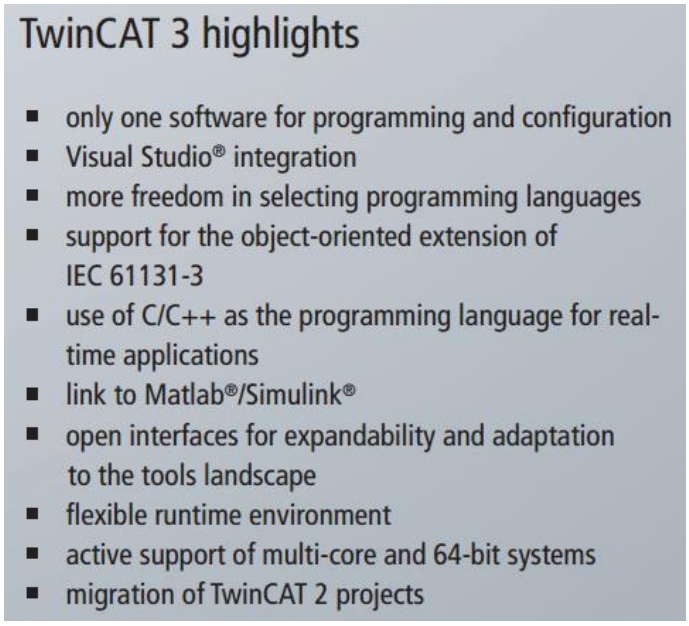


Figure 20 TwinCAT 3 main features (Beckhoff 2012b).

The main philosophy behind TwinCAT 3 is modularity. Each system or function is represented in TwinCAT as modules and therefore independent from other control functions in the system. With this in mind, it is possible to integrate a large number of different modules on the same system, communicating among them using a standard, language independent transport layer. In TwinCAT, the transport layer is called ADS which stands for Automation Device Specification. Figure 21 presents TwinCAT 3 runtime's architecture. As a result of its modularity, TwinCAT 3 offers effective multi-core support by using different cores for different modules or tasks. The distribution of tasks and modules among the processor's cores can be done either automatically by the system or defined by the user. Figure 22 presents a graphical explanation of multi-core support in TwinCAT 3.

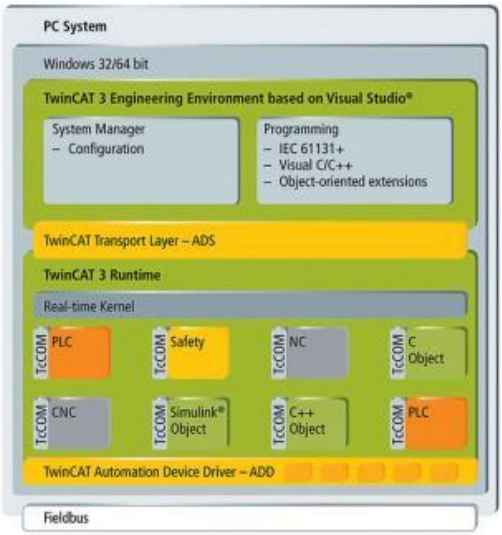


Figure 21 Graphical representation of TwinCAT3's runtime architecture (Beckhoff 2012b).

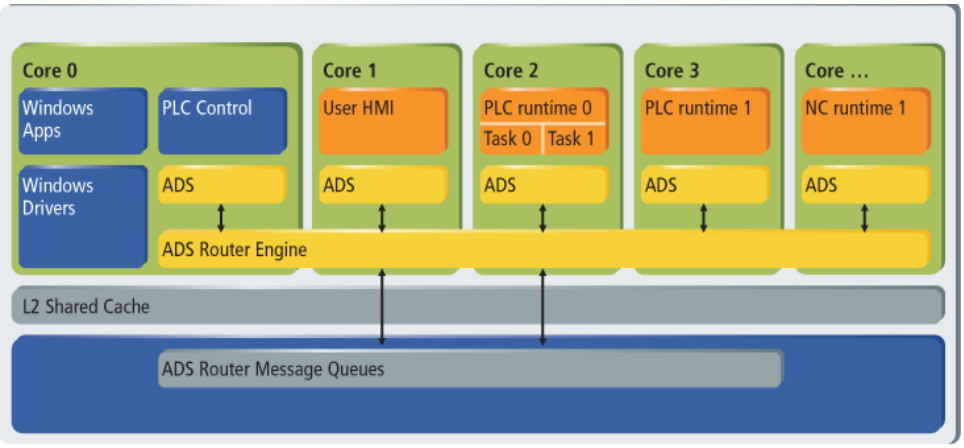


Figure 22 Multi-core support in TwinCAT 3 (Beckhoff 2012b).

It is worth mentioning that while TwinCAT 2 can be installed as a standalone program, TwinCAT 3's programming environment is integrated in Microsoft's Visual Studio (Figure 23). This feature should not represent any inconvenience for the user, since the basic Visual Studio shell is included together with the TwinCAT 3 installer. Furthermore, the integration of all the modules in TwinCAT under the same framework presents a number of benefits including handling, connection to source code control software, standardization, debugging.

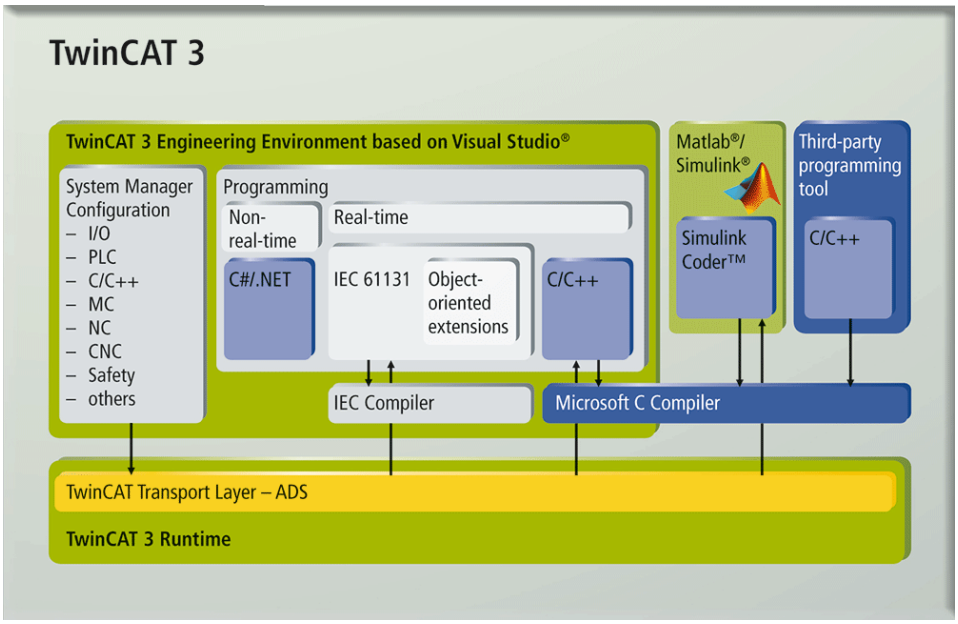


Figure 23 TwinCAT 3 environment and runtime architecture (Beckhoff 2012b).

Two of the most important features included with the new version are the native support for C/C++ programming language and integration of the Matlab/Simulink. C/C++ language is a standardized, widely used and powerful programming language and by integrating it directly into the TwinCAT environment the door to countless new and old projects has been opened. On the other side, Matlab/Simulink is widely used in the scientific and research environments.

2.6.1 Motion control with TwinCAT 3

TwinCAT 3 offers simple, yet powerful solutions for motion control. From simple Point-To-Point (PTP) movements to the most demanding robot applications, all are possible under the same system. Figure 24 displays the motion control functionality available in TwinCAT 3.

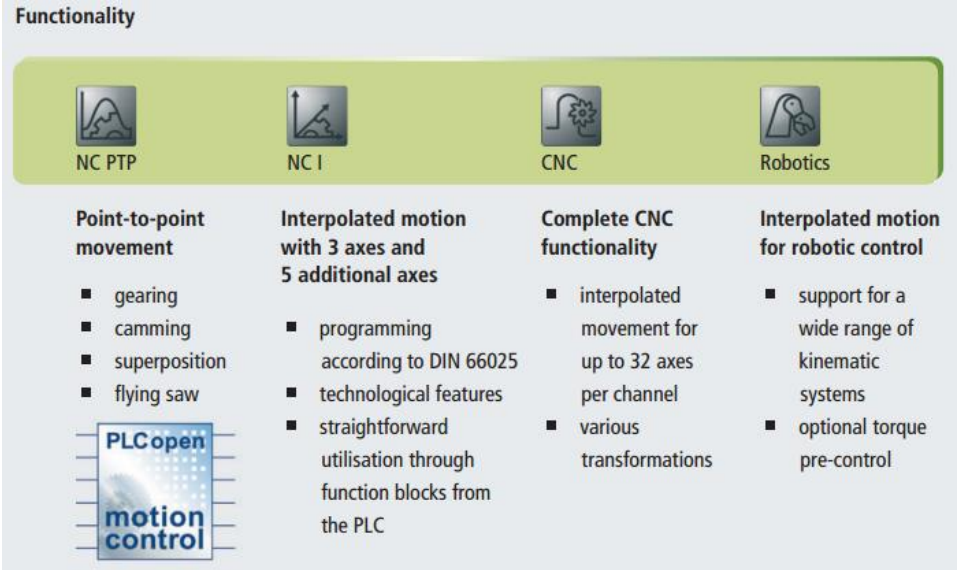


Figure 24 Motion Control functionality modules available in TwinCAT 3 (Beckhoff 2012b).

It is possible to access and control motion functions inclusive from the PLC module, allowing the user to design personalized motion systems and HMIs (Human Machine Interfaces).

Depending on the end application, a different functionality will be taken into use. For example, in an automatic drilling system it is enough to implement Point-To-Point movements, while in the case of 3D printing or milling, interpolated motion must be implemented.

3 MACHINE DESIGN

“The first step in technical design requires a paper, a pencil and a steady hand” (Marcos 2010).

Following a consideration of the advantages and disadvantages of CNC machines and 3D printers presented in the introduction chapter, a first draft was sketched, as illustrated in Figure 25.

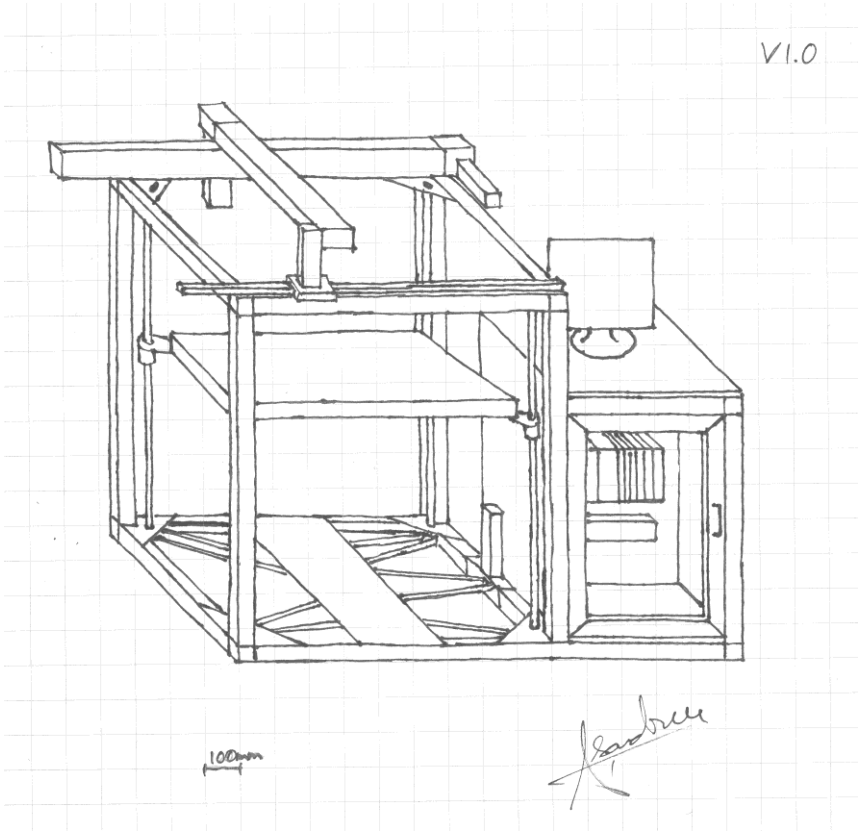


Figure 25 First sketch of the 3D-Workbench concept.

After some deliberations regarding material's strength and stability, it was decided to use aluminium profiles with increased rigidity to reduce elasticity in the frame, as well as two linear modules for the X-axis movement (increased accuracy). A concern pointed out by Mr. Väisänen regarding industry standards was taken into account as well: an industry standard, pre-assembled cabinet replaced the cabinet made out of aluminium profiles. In addition, a shorter circuit for the belt of the Z-axis was designed, reducing possible elastic movement and therefore inaccuracies (Uotila 2015). Connecting a motor directly to one ball-screw would decrease mechanical inaccuracies and hold the platform in case of breakage of the belt. Still, without a proper system for detecting a breakage the Z-axis motor might continue rotating one axis while the other three axes remain fixed, leading to a bending of the axes or even mechanical ruptures. Next, a second draft was sketched, presented in Figure 26.

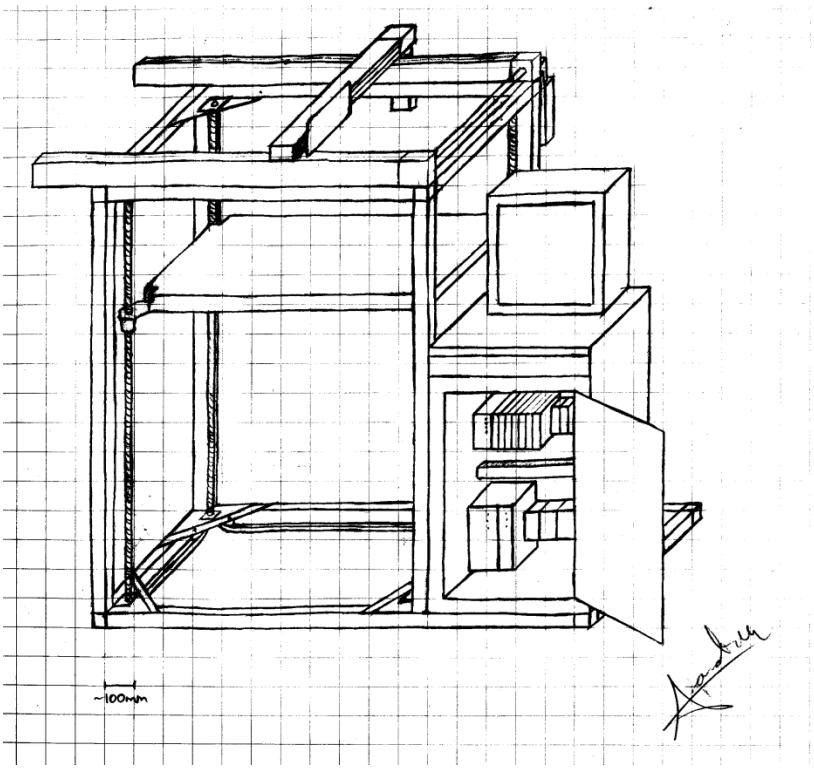


Figure 26 Second sketch of the 3D-Workbench concept.

The second sketch provides an overall view to the main components needed, as well as their placement. The selection process for the most important components is detailed in the next chapter.

3.1 Components selection

3.1.1 Metallic frame

Similar to a consumer 3D printer's frame, the metallic skeleton of the machine developed in this thesis project was made of aluminium profiles, in contrast to steel structures used in large industrial CNC machines. The reason behind this choice was based on the mechanical properties of aluminium, which make it much easier to work with. In addition, aluminium has a much lower density compared to steel, and therefore the overall weight of the machine could be reduced.

Bosch-Rexroth (2015) offers a wide range of aluminium profiles in their online shop for light to heavy-duty applications. Each type (called series) includes different optional accessories for structural mounting. The "60x60H" profile (Figure 27) inside the "60 Series Profiles" category offers increased strength and structural rigidity, which makes it the ideal choice. The "60" indicates the profile's width (60mm), while "H" stands for "Hardened" (increased strength for heavy-duty applications).

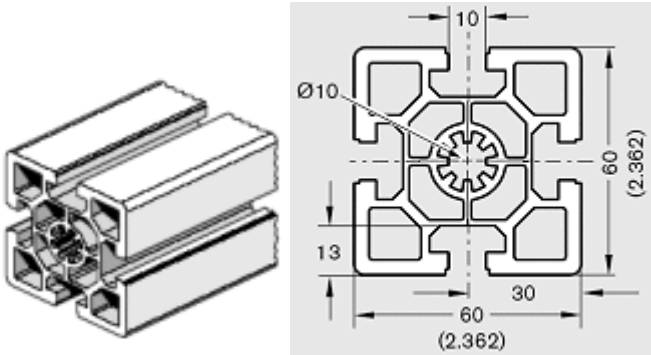


Figure 27 Two views of the 60x60H aluminium profile (Bosch Rexroth 2015).

3.1.2 Enclosure cabinet

Enclosure cabinets are the appropriate solution for providing a high degree of protection against external elements for the PLC and the other control components. When choosing a suitable cabinet, the following requirements were set: it should provide enough space to fit the PLC controller, additional modules and the power supply, control drives, relay modules; and at least a protection grade IP60 (no ingress of dust allowed). Nevertheless, the protection grade may be changed according to customers' needs.

After having examined the products offered by three different companies (Eldon, Fibox and Rittal), it was found that the most suitable solution was offered by Rittal. Their industrial workstation with model number 6901.100 met and even surpassed the requirements set.

After downloading Rittal's own 3D modelling software, RiCad 3D including their catalogue, it was possible to obtain a 3D model of the required enclosure (Figure 28) which was later exported to Autodesk Inventor.

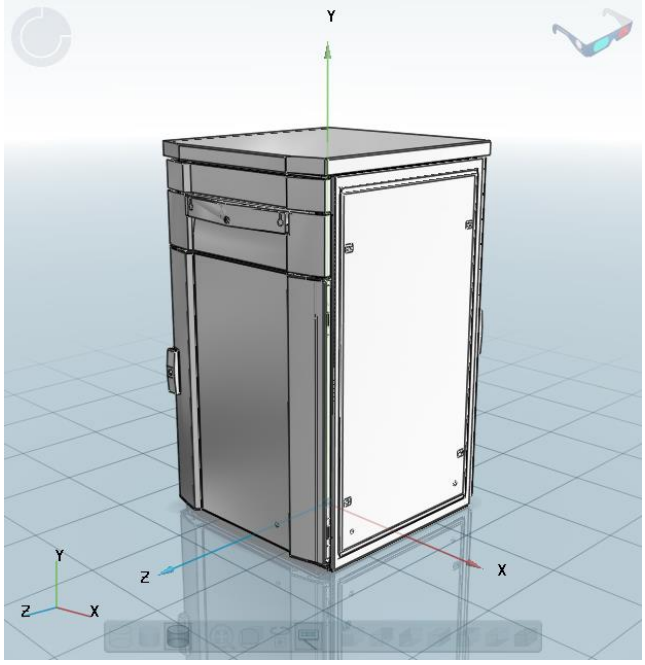


Figure 28 Rittal enclosure cabinet model 6901.100 (Rittal 2014).

3.1.3 Linear modules

There are many different types of solutions for implementing motion available on the market as it was pointed out in the chapter on servomotors and drives. In addition, for these systems to produce a linear motion (except for linear motors and pneumatic drives); they must be coupled with linear modules. For the system presented in this thesis, the following requirements were established: for the X and Y-axis (horizontal movements), high dynamic capacity together with accuracy and repeatability were a must; as for the Z-axis (vertical movement), dynamics were not as important as load capacity and positional accuracy.

Examples of linear modules include a belt, ball screw, or rack and pinion driven system. When choosing the linear motion system, it is important to consider the whole system, not only the particular task of each component. There are seven key factors that must be taken into consideration when selecting linear products and can be easily remembered with the acronym “LOSTPED”: Load (force the system must apply and withstand), Orientation (or plane of travel, determines the direction of the force), Speed (including acceleration, impacts actual loads for linear bearings and drives), Travel (determined by two times the stroke length times the total number of cycles expected before replacement of the motion component), Precision (includes travel accuracy, final positioning and repeatability), Environmental (temperature and dirt, impact linear motion design) and Duty cycle (active time of the system, affects the heating of the motor and other motion components). (Bosch-Rexroth 2014.)

Based on the above and the requirements specifications of the system developed in this thesis project, the following was determined: for the X and

Y-axis, a toothed belt driven system would be the best solution; while for the Z-axis, a ball screw driven system would be more suitable.

Among the toothed belt motion systems offered by Rexroth, the one that best fitted the needs as to dynamics, accuracy, repeatability and cost was the *Linear Module with Ball Rail System and Toothed Belt Drive (MKR)*, which offers a repeatability of up to 0.1 mm and lead constant of 110 mm/turn. For the X-axis movement, two MKR linear modules are driven synchronously with one single servomotor through a shaft connecting both. The motor is connected to one of the linear modules through a 90º angle gearbox unit. Regarding the Y-axis, it is sufficient to use a single MKR module lying on the carriages of the X-axis modules. This configuration is based on the model offered on Rexroth's webpage, as shown in Figure 29.

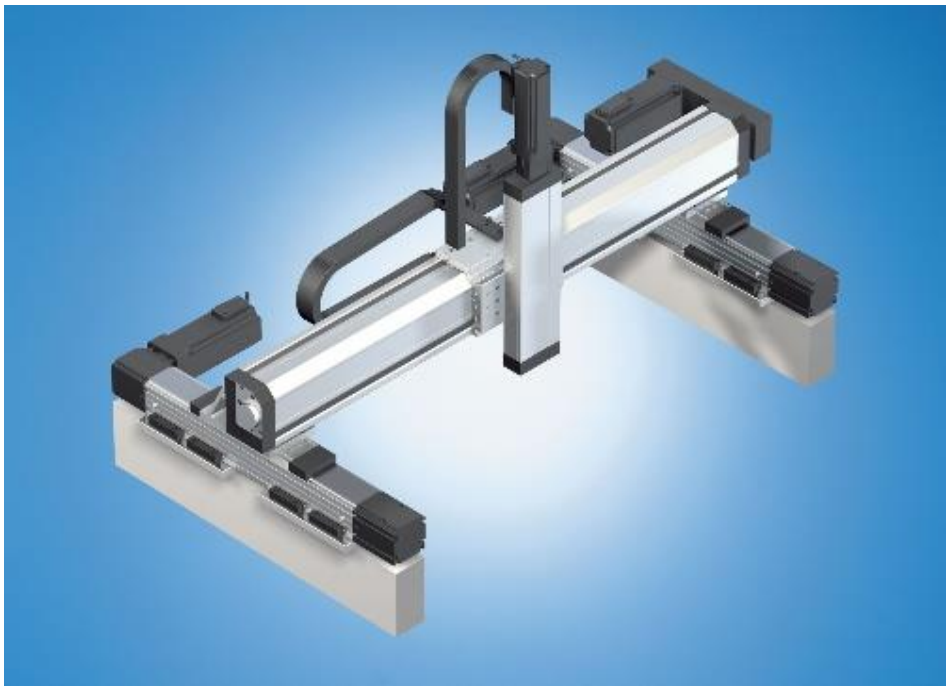


Figure 29 Bosch Rexroth bridge module H (Rexroth n.d.).

For the Z-axis movement, the product which best fulfilled the need for a compact size, accuracy and load capacity was the *ECOplus Series with Recirculation Caps*. Among its characteristics, it is worth mentioning industry-leading speeds (up to 150 m/min) and a high lead accuracy (T5, 0.023/300mm). (Bosch-Rexroth 2014.)

3.1.4 Motors and servo drives

The first step when sizing motors is to determine the appropriate maximum power of the motor to develop the required task. On one hand, if the motor is sized incorrectly it may not have sufficient torque to accomplish the tasks given or at least not in an acceptable manner. On the other hand, the price of motors increases with their rated power and therefore it can be an expensive mistake to use extra-large motors.

The motor dimensioning for the axes was conducted by Mr. Uotila from Beckhoff, using specialized software called “Neugart Calculation Program”. Beckhoff was provided with the following data in order to perform the calculations:

- For the X-axis movement: 17 kg mass to be moved horizontally (including the Y-axis linear module, motor and attached tool), movement using AT5 toothed timing belt, 37 mm diameter driving wheel with 24 teeth, 900 mm maximum stroke, 0.4 m/s maximum speed and high dynamics capabilities.
- For the Y-axis movement: 6 kg mass to be moved horizontally (including carriage of the linear module and tool attached), movement using AT5 toothed timing belt, 37 mm diameter driving wheel with 24 teeth, 500 mm maximum stroke, 0.4 m/s maximum speed and high dynamics capabilities.
- For the Z-axis movement: 100 kg mass to be lifted vertically (including the mass of the platform and item to be printed, drilled or milled), four ball nut screws used with 20 mm diameter and 5 mm lead, 1000 mm maximum stroke, 0.2 m/s maximum speed and internal brake fitted from victory.

Because of the requirements specified, it was necessary to include a gear box for the X and Y-axis movements in order to keep the inertia ratio as close as possible to one (Uotila 2015). A graphical presentation of the dynamics requirements expected in the X, Y and Z-axis can be found in the Appendix 1.

Based on the requirements set for each axis, it was recommended to use the Beckhoff AM8042 motor (characteristics can be seen in Table 2) together with a PLE60 gear box for the X and Y-axis movements, and the Beckhoff AM8032 motor (characteristics in Table 3) for the Z-axis movement.

Table 2 Beckhoff AM8042-wFy mechanical characteristics (Beckhoff 2014b).

Data for 400 VAC	**AM8042-wFy**
Standstill torque	4.10 Nm
Rated torque	3.70 Nm
Rated speed	5000 min^{-1}
Rated power	1.94 kW
Peak torque	18.5 Nm
Standstill current	4.10 A
Peak current	22.7 A
Number of poles	8

Rotor moment of inertia	1.97 $kgcm^2$
Weight	4.7 kg

Table 3 Beckhoff AM8032-wEy mechanical characteristics (Beckhoff 2014c).

Data for 400 VAC	**AM8032-wEy**
Standstill torque	2.37 Nm
Rated torque	2.20 Nm
Rated speed	6000 min^{-1}
Rated power	1.38 kW
Peak torque	11.66 Nm
Standstill current	2.95 A
Peak current	17.2 A
Number of poles	8
Rotor moment of inertia	0.842 $kgcm^2$
Weight	2.8 kg
Holding torque brake	2.0 Nm

The servo drives recommended by Beckhoff were based on the power requirements of the motors. For the X-and Y-axis motors, a recommended servo drive was AX5203 model. This particular model offers the possibility to independently control two servo motors, thus reducing space and costs. For the Z-axis, the commended servo drive was the single channel AX5103 drive. (Uotila 2015).

Regarding the position sensors, both models use Beckhoff's One Cable Technology with 18 bits of resolution. Combining both systems (the linear modules and the motors), the following results regarding accuracy and repeatability were obtained:

- For the X and Y axis: the maximum theoretical accuracy can be determined by dividing the lead constant (110 mm/turn) by the position sensor's resolution (2^{18}-1 increments/turn) and therefore it equals to a control resolution of 0.0004 mm; while the repeatability was directly obtained from the technical datasheet of the linear modules and equals to 0.1 mm. Therefore, the absolute theoretical accuracy is 0.0002 mm ±0.1 mm.
- For the Z axis: the accuracy was determined by combining the accuracy of the linear modules with the accuracy of the motor. The absolute theoretical accuracy is 0.0004 mm ±0.023 mm.

3.1.5 Controller and I/O cards

Even though with TwinCAT installed, almost any personal computer can act as a Programmable Logic Controller, the solution for the controller was based on an industrial grade CPU unit from Beckhoff. On one hand, the decision was based on the functional stability present in Beckhoff systems, since they are designed for harsh environments. On the other hand, combining PC functionalities with PLC interface for modules in the same enclosure reduces power consumption and costs, and increases efficiency in space management.

Using the expert advice offered at Beckhoff, it was found that a suitable system would be the embedded PC series CX5120 (Figure 30), which has the smallest CPU capable of interpolation movements. (Uotila 2015.)

The CX5120 is powerful enough for basic operations, which include milling, 3D printing or drilling among others. In case the customer requires higher performance, for example to run third-party demanding software, a different model may be used such as the CX5130 or CX5140.

In addition, the CX5100 family may include optional interfaces (such as RS232, PROFIBUS, CANopen, EtherCAT), further expanding the interconnection with customer defined tools. The operating system (optional) for the CX5120 CPU is Microsoft Windows Embedded Standard 7 P. (Beckhoff 2015a, 8.)

Figure 30 Top view of the CX5120 CPU from Beckhoff (Beckhoff 2015b)

Technical data about the CX 5120, CX5130 and CX5140 can be found in Appendix 2.

Since the controller is completely modular, only its basic functions are detailed next. In case additional features are required, they can easily be implemented by adding the correct terminal. As to the basic functions, the system is expected to:

– Control the servo drives.

- Read the information from sensors (six in total).
- Detect when the emergency stop button is pressed.
- Power on and off a simple tool.
- Open or cut the pressure air supply to a tool.

Based on the above, it was determined that the required terminals for the controller were as follows:

- One EtherCAT extension terminal model EK1110.
- One, 8-channel digital input terminal (24 V DC) model EL1008.
- One, 8-channel digital output terminal (24 V DC) model EL2008.

3.1.6 Additional components

In this chapter, a brief description is given on smaller functional elements needed for providing the basic functions.

In addition to the controller and terminals, a regulated 24 V power supply was needed. The current rating was obtained from the documentation of the controller, which states that to ensure a correct operation of the CPU and attached terminals, “the power supply must supply 4 A at 24 V” (Beckhoff 2015a, 32).

The 8-channel output card might not always be able to provide sufficient current and therefore, four relays were used to ensure a safe operation for the 24 V logic operations and one additional relay capable of driving 220 V for controlling certain tools (such as the drilling or milling motor).

Pressurized air was also a requirement for certain tasks and therefore control elements were included, such as main manual switch for air supply and a 24 V driven switch for the controller to operate the tool. Both switches should be able to withstand at least 6 bar pressure.

The components’ model and manufacturer are not specified, since they can easily be changed to other models or manufacturers based on availability, price or other reasons. For this project, generic components were used for the reader to obtain a visual distribution of components in the 3D Modelling chapter.

3.2 3D Modelling

After all the components were determined, a 3D model of them was needed. Although in most cases the manufacturers offer 3D models of their products, this is not always the case. To overcome this, manufacturers always offer technical drawings of their products, which can be used to produce a 3D model. In almost all cases when using a ready-made 3D model, it is necessary to first open the file using the command *Open* in Autodesk Inventor and then save it in the same or different location. This creates an updated file which can easily be imported to other Inventor projects. Some parts were tailored made to suit the present project and their technical drawings can be found in the Appendix 3 of the present thesis.

Because CNC machines require high dynamics while following an interpolated path, it is important to design a structure as rigid as possible, avoiding any type of “elastic” movements which could interfere with the control of the servomotors causing undesirable oscillations and therefore reducing the performance of the machine. The structure used for the metallic body of the machine is similar to a cage, with reinforcements on the sides. This type of structure offers great stability, hardened rigidity and small print (i.e. occupies limited volume space). The base has been extended to accommodate the cabinet containing the motor drives, PLC and other electronics. Figure 31 depicts the basic metallic skeleton of the 3D-Workbench including reinforcements. Regarding the dimensions, these have been calculated taking into account the desired working area, which for a “standard” version of the 3D-Workbench system is expected to be: 900x500x1000 (LxWxH).

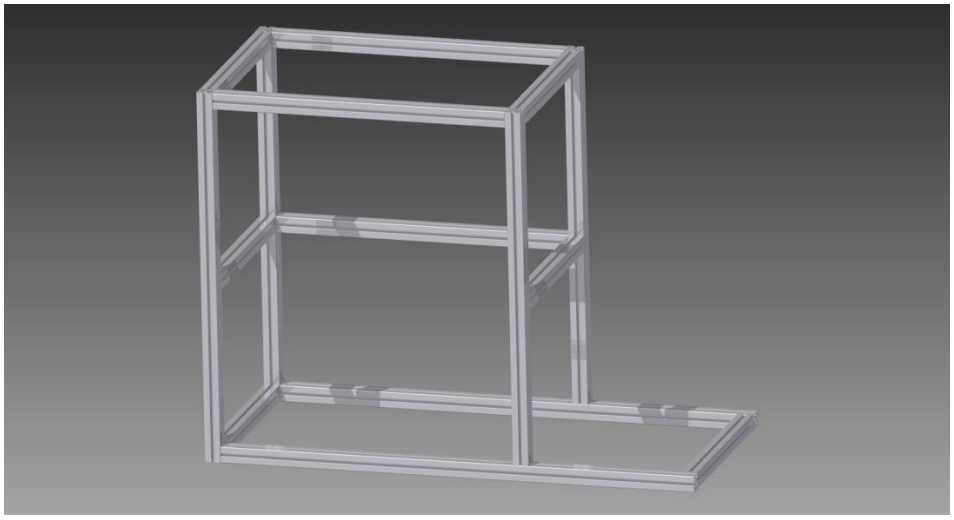


Figure 31 Basic metallic structure design in Autodesk Inventor.

In order to improve stability and load capacity a total of four ball screw assemblies were used; one in each corner of the platform, as can be seen in Figure 32. The assembly of the four ball screws with the metallic frame was done using triangular shaped, steel metal plates. Because of the stress endured by the triangular plates, it became obvious the need for a reinforced material such as steel.

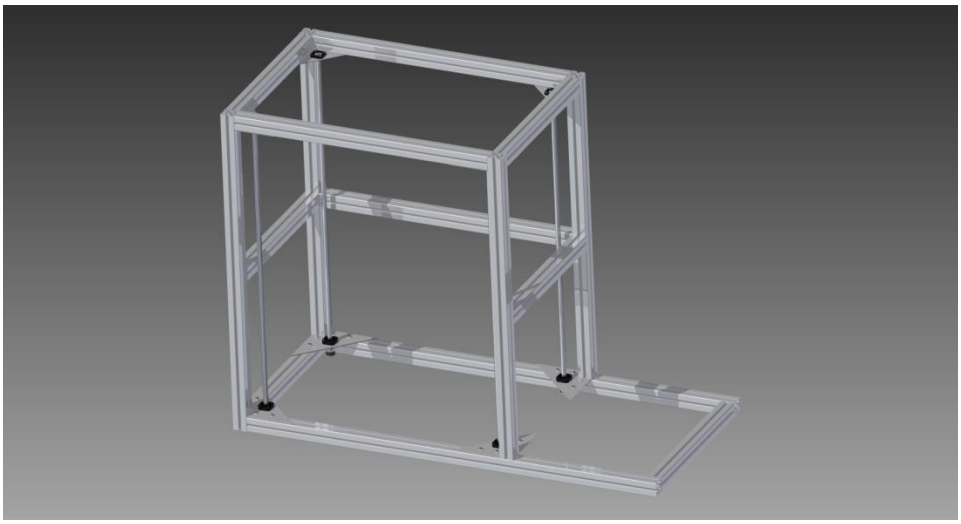


Figure 32 Ball screw assemblies mounted on the metallic frame.

The connection between the four ball screw assemblies was done using AT5, 16mm wide synchronous belt. The connection between the belt, the ball screws and the motor was done using exclusive parts designed in Autodesk Inventor and presented in the Appendix 3. In order to tension the belt, two screws must be adjusted, as can be seen in Figure 33.

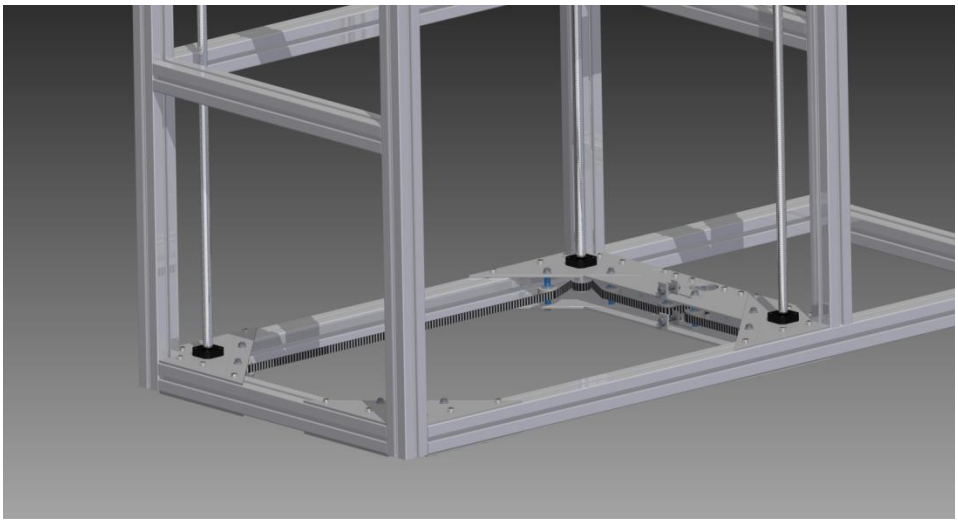


Figure 33 Z-axis movement assembly.

Once the structure sustaining all the components was ready, the next step was to design the working platform in the Z-axis. The main characteristic of the platform is that it must provide sufficient holding strength with minimal bending, while at the same time maintain a reduced weight. The material used for the frame of the platform was again aluminium profiles with hardened structure. Because different applications have different requirements, the platform was divided in two main types. The first one is oriented to applications such as milling, drilling or stamping; and is designed to offer a firm grip of differently shaped parts made of diverse materials. Grips in the shape of an “L” are attached to the platform and tightened to a part, holding it firmly. The other type of platform is specialized for the tasks of 3D printing and includes a heating resistance under a glass sheet.

In the printing process, it is important to heat the platform for the base of the part being printed not to cool down. If this happens, the difference in temperature between the upper part of the component and the lower part would deform the component, causing undesired results. The temperature of the printing platform ("bed") can be regulated from the PLC through an analog output (additional terminal not used for the present example project). Figure 34 presents the two platform types in use.

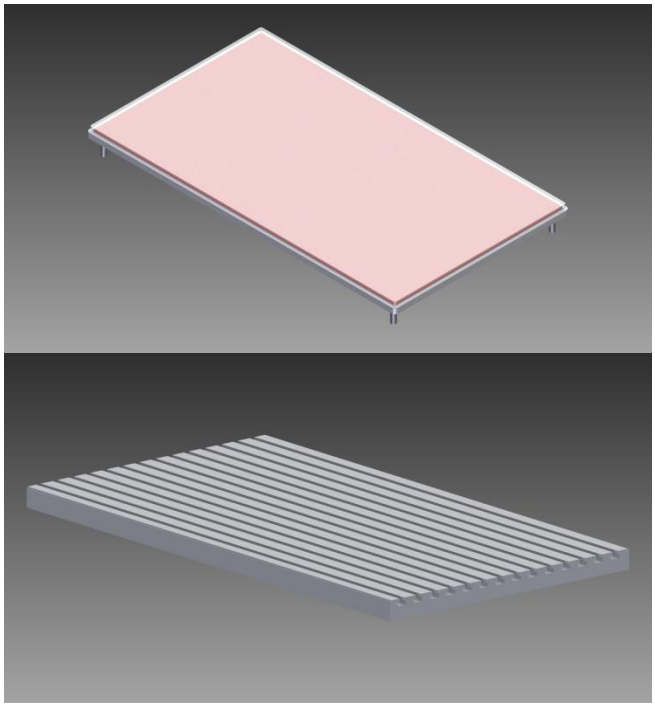


Figure 34 Platform for printing (top) and for milling, drilling and stamping (bottom).

Next, the H-bridge setup for the X and Y-axis movement was assembled over the metallic structure using 3D models obtained from Rexroth's webpage. Then, the motors for X, Y and Z-axis were attached using 3D models obtained from Beckhoff's webpage.

The previously exported cabinet model from RiCAD 3D program was imported into Autodesk Inventor and placed in its corresponding place on the metallic structure. Figure 35 presents the updated model of the project up to this point.

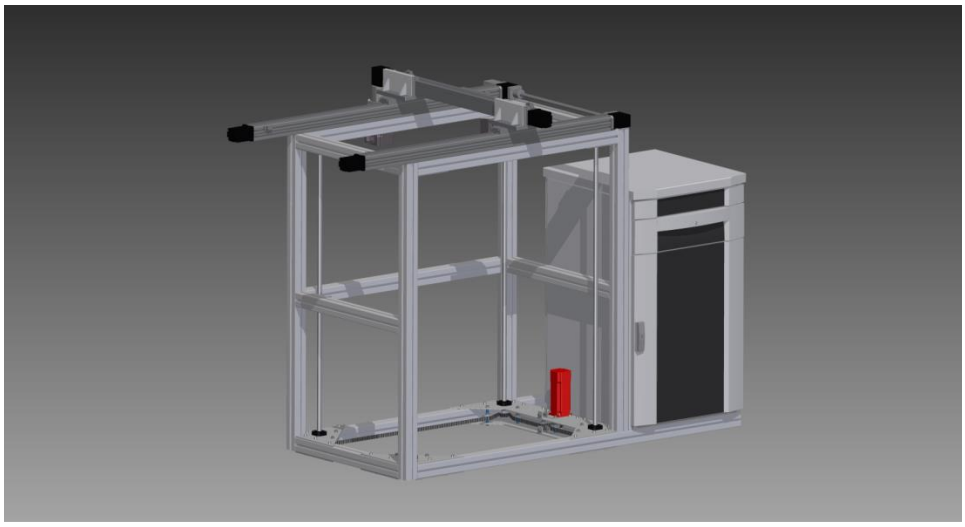


Figure 35 Model including frame, linear modules, motors and cabinet.

The next step was to install the components inside the cabinet. The ready-made model did not include a rack and therefore it was created as a new part. Figure 36 presents a view of the cabinet with all the components installed. In addition to the components, organizing racks for cable management were created as well.

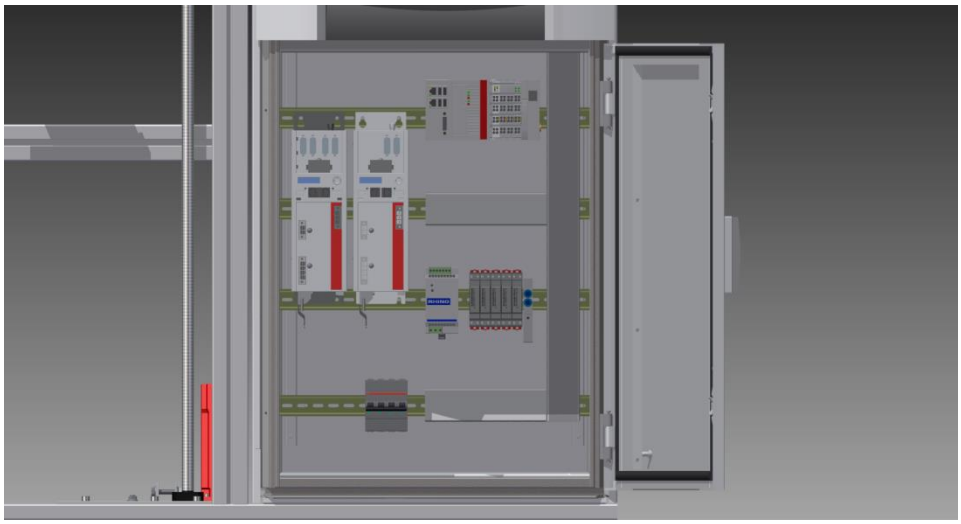


Figure 36 Front view of the cabinet with components installed.

3.3 Electrical installations

A general view of the electrical connections between the components can be seen in Figure 37.

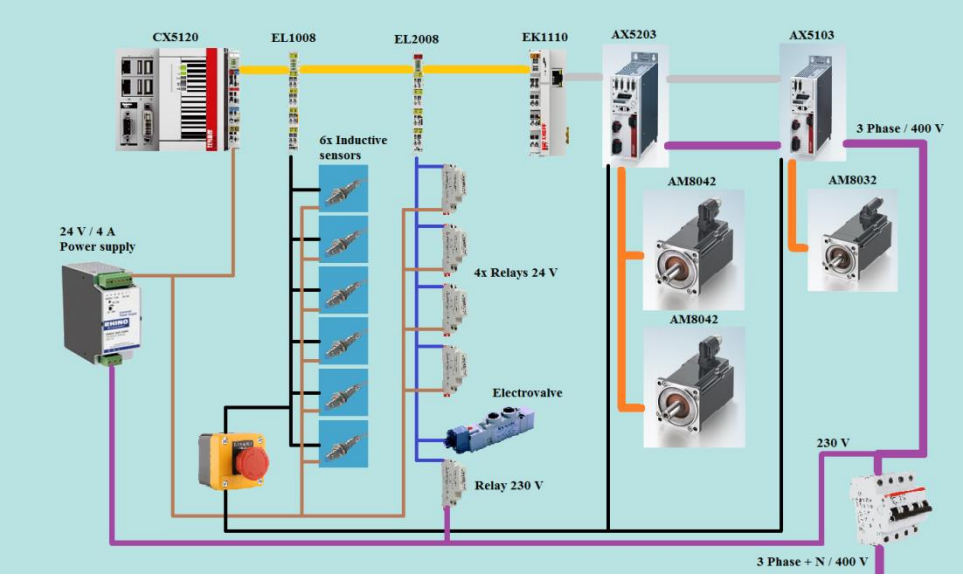


Figure 37 Connections among the components.

In addition, a technical diagram of the front view of the cabinet was exported from the three dimensional model in Inventor. This diagram can be found in the Appendix 4.

3.4 Configuration and programming

3.4.1 TwinCAT NC configuration

As it was mentioned before, in TwinCAT the module taking care of movements is called NC Motion. Next, a brief guide on how the NC module should be configured is presented. Certain configuration parameters can only be adjusted while configuring a physical system. These parameters are only explained, but not configured.

The first step consisted in implementing a new motion configuration. In this case, the motion configuration type was *NC/PTP NCI Configuration*. Two tasks are created under the motion module: SAF and SVB. The SVB task is in charge of generating the velocity and position control profiles respect to the current position for each of the axis; while the SAF task is in charge of sending the information generated in the SVB task to the drive. The SAF task should be configured to run five times faster than the SVB task. The default control loop is presented in Figure 38. (Infoplc n.d.)

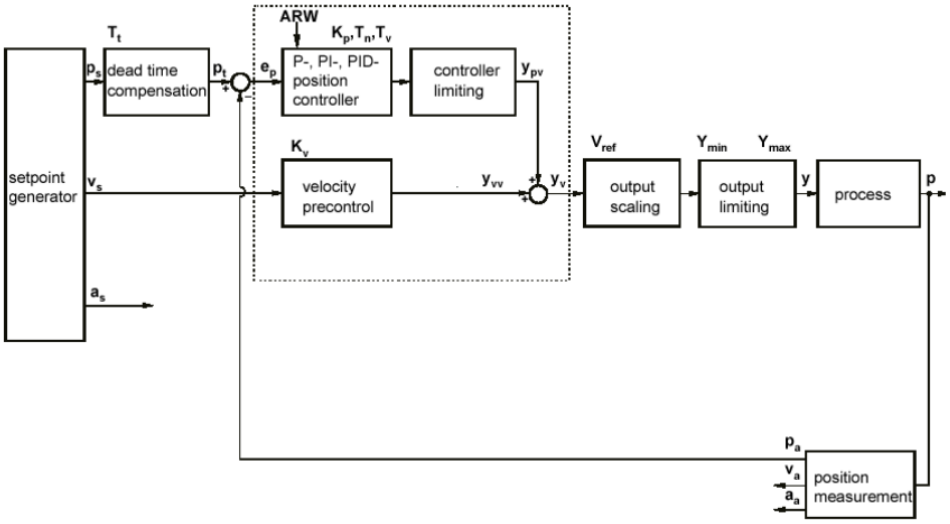


Figure 38 Default NC control loop in TwinCAT (Infoplc n.d.)

The next step was to add three axes by right clicking on the tab *Axes* and selecting *Add new item*. The result can be seen in Figure 39.

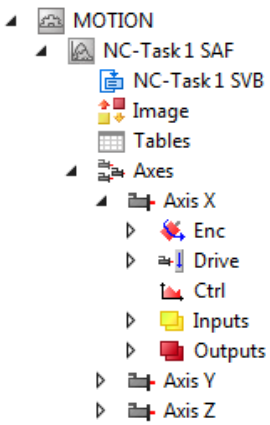


Figure 39 Adding axes to the system.

As can be seen from Figure 39, under each axis the following parameters can be found: Enc (Encoder configuration), Drive (drive configuration), Ctrl (control loop configuration), Inputs and Outputs (input and output variables associated with the selected axis).

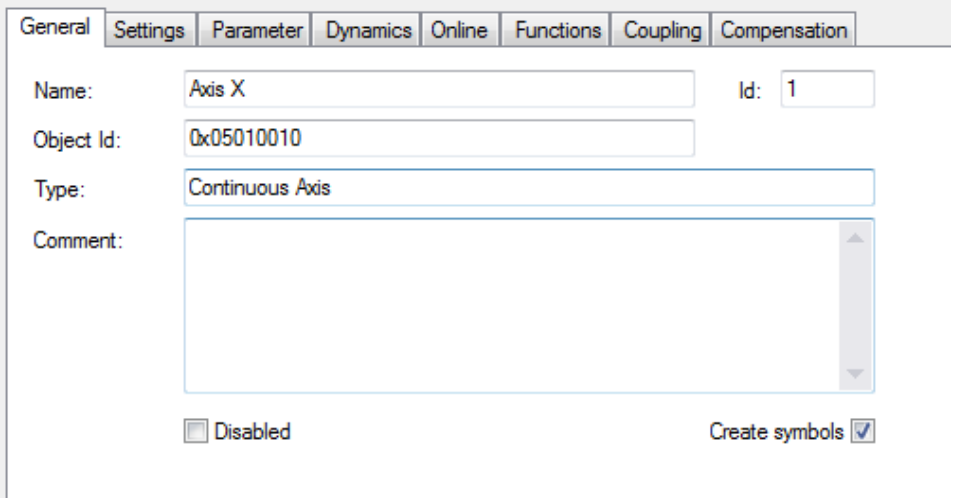


Figure 40 Axis general properties.

When double-clicking on one axis, the general configuration window is opened for that particular axis (Figure 40). In the *General* tab, if *Create symbols* is selected, the system will create table entries of the variables. This can be convenient for example when Scope View is used. In the *Settings* tab the axis type can be selected, as well as the units used. For the present project, a digital communication using EtherCAT was selected.

Among the options present in the *Parameter* tab, the most relevant ones are *Velocities and Monitoring*. Under *Velocities*, the motor's maximum velocity must be input under *Reference velocity* (this refers to the output speed of the motor when commanded to run at maximum speed). *Maximum* velocity limits the output of the drive, not allowing the motor to surpass a predefined maximum velocity. Manual *Velocity* determines the maximum speed when the drive is manually controlled (i.e. jog). *Calibration Velocity* determines the homing velocity. Under *Monitoring* it is possible to set certain monitoring tasks, such as position lag, which can increase the security of the system (for example, if external forces prevent the motion the system will stop and issue an error). In addition, in case G-Code is used the parameters under *NCI Parameter* must be configured. These parameters are *Rapid Traverse Velocity (G0)* which refers to the maximum speed of the system while positioning for a task and *Velo Jump Factor* which refers to the reduction factor of the velocity in curves.

Next in the *Dynamics* tab, the acceleration and deceleration profiles can be set either by directly inputting the values or by letting the system calculate the values according to specified acceleration and deceleration times.

In the *Online* tab an overview of the state of the selected axis is presented. Also in this tab it is possible to manually jog the motor (through the F1, F2, F3 and F4 buttons), command the motor to certain position (F5) or even perform a homing command (F9) to calibrate the axis. Before being able to control the motor, it is necessary to enable the controller by pressing the *Set* button and then enable the controller, the required rotating direction and an override value greater than zero.

The *Functions* tab includes predefined functions which can be used to test or monitor the axis. The function *Reverse sequence* can be used while configuring the controller, for it moves the axis from one set point to another and then back to the previous point at a predefined speed.

In the *Enc* subsection of the axis (Figure 41), the encoder type used can be defined, together with configuration values. In the *NC-Encoder*, the encoder type can be selected and linked to the corresponding encoder terminal. Under the *Parameter* tab, the following parameters should be changed according to the system's characteristics: *Invert Encoder Counting Direction* (depends which direction is considered to be positive), *Scaling Factor Numerator*, *Reference System* (what type of data does the encoder provide), *Invert Direction for Calibration Cam Search* (while homing the axis) and *Calibration Value* (position the drive should adopt after homing).

Scaling Factor Numerator is an important parameter which determines the travelled distance of the axis per increment of the encoder. Thus two values must be determined: the encoder's accuracy (increments per revolution) and the travelled distance of the axis per one revolution of the motor. Once these two values are obtained, dividing the travelled distance per revolution by the increments per revolution presents as a result the scaling factor. (InfoPLC n.d.)

On the *Online* tab an overview of the current state of the encoder is presented. In addition, it is possible to manually calibrate the axis by inputting an absolute position for the axis and setting the calibration flag.

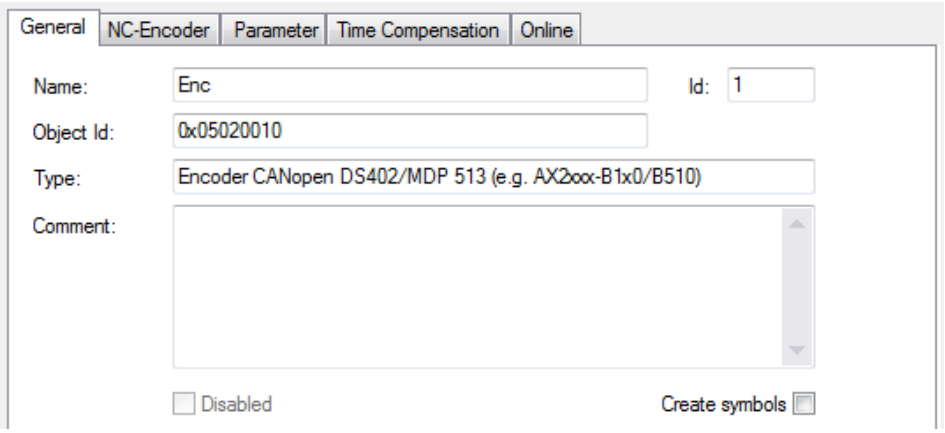


Figure 41 *Enc* subsection of an axis.

On the *Drive* subsection, the servo drive can be configured. Among the configurable parameters, the most important can be found under *Output Settings* in the *Parameter* tab (Figure 42). *Invert Motor Polarity* inverts the rotating reference for the NC Motion system and *Reference Velocity* is the same parameter as in the general configuration of the axis.

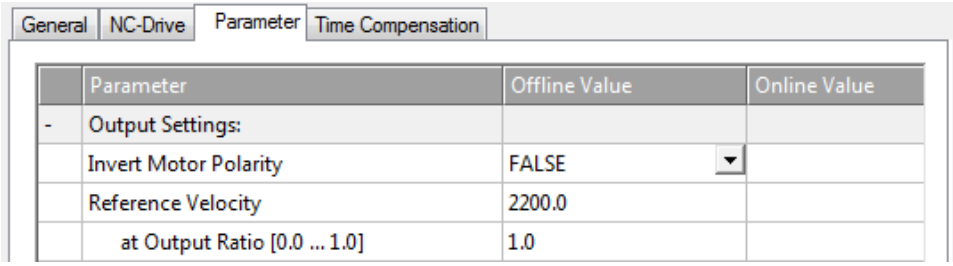


Figure 42 Servo drive configuration.

Lastly, on the *Ctrl* subsection (Figure 43) the controller type and controller's parameters can be adjusted. On the *NC-Controller* tab the controller type can be selected. In most systems, it is enough to use *Position controller P*, although if the expected control cannot be obtained, a PID control can be selected. In case the *Position controller P* is selected, on the *Parameter* tab the *Proportional Factor Kv* and *Pre-Control Weighting* should be adjusted.

One way for determining the Kv-factor is to drive the axis between two points at a constant speed, while increasing the Kv-factor. Once the system starts to oscillate while driving at constant speed, the Kv-factor should be decreased by 30%. (InfoPLC n.d.)

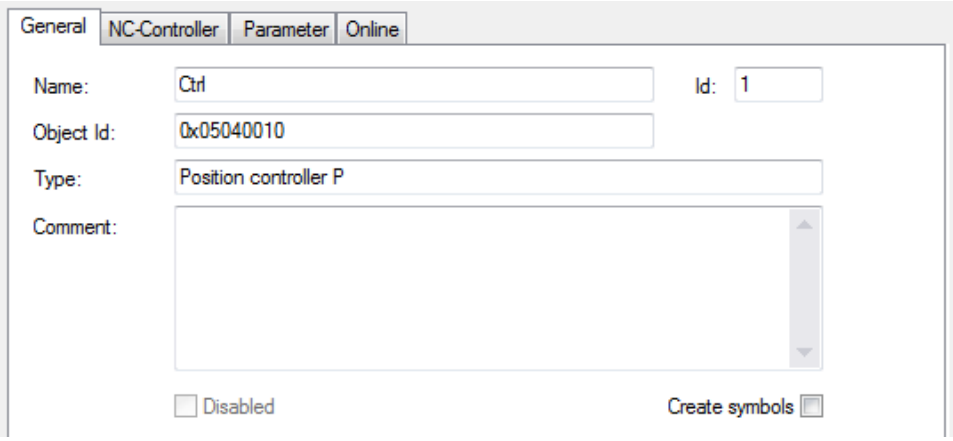


Figure 43 *Ctrl* subsection present under an expanded axis.

After all the axes were correctly configured, the next step was to add an interpolation channel which can be used for G-code commands for example. Adding an interpolation channel is done by right clicking on the *NC-Task* and then selecting *Add new item*. A new window appears (Figure 44), where the type should be *NC Channel (for interpolation)* and the name can be user defined.

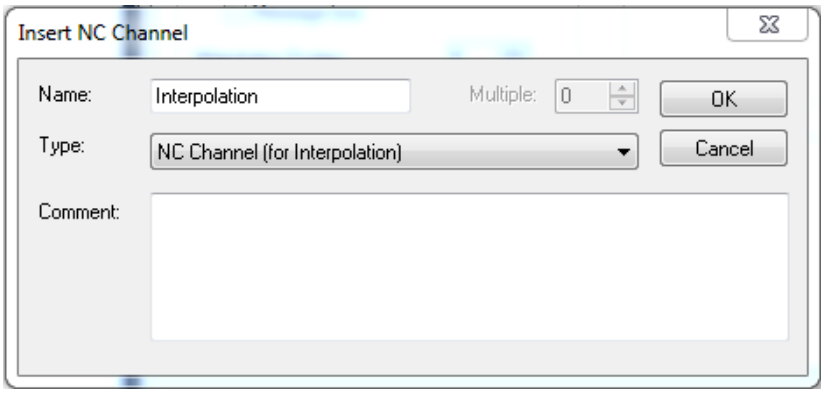


Figure 44 Window for creating a new interpolation channel.

After clicking *OK*, the motion module should look similar to Figure 45.

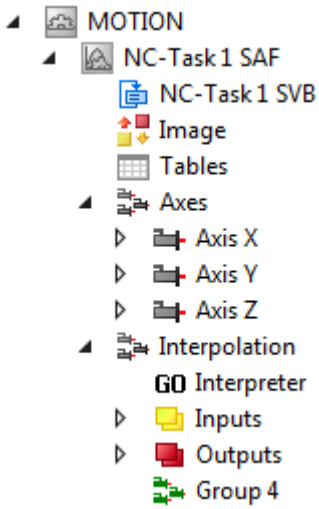


Figure 45 Motion module and subsystems.

Before being able to use G-code, the system must be set up. The following process should be done every time TwinCAT is restarted.

After the axes were correctly set up and adjusted, it is necessary to activate the configuration and restart TwinCAT in *Run mode*. Once in *Run mode*, the axes need to be assigned to the interpolation group. This is done by navigating to the group interpolation created in the previous chapter (*Group 4* in this example) and there on the *3D-Online* tab (Figure 46). TwinCAT organizes by default the axis under *Nominal assignment*, although this can easily be changed. *Q1* to *Q5* refer to the slave axis (not used in the present project). Once the right axis is selected for each coordinate vector, *Accept Assignment* must be pressed. On the *Online* tab the motion module status is displayed together with the error codes (if any) present.

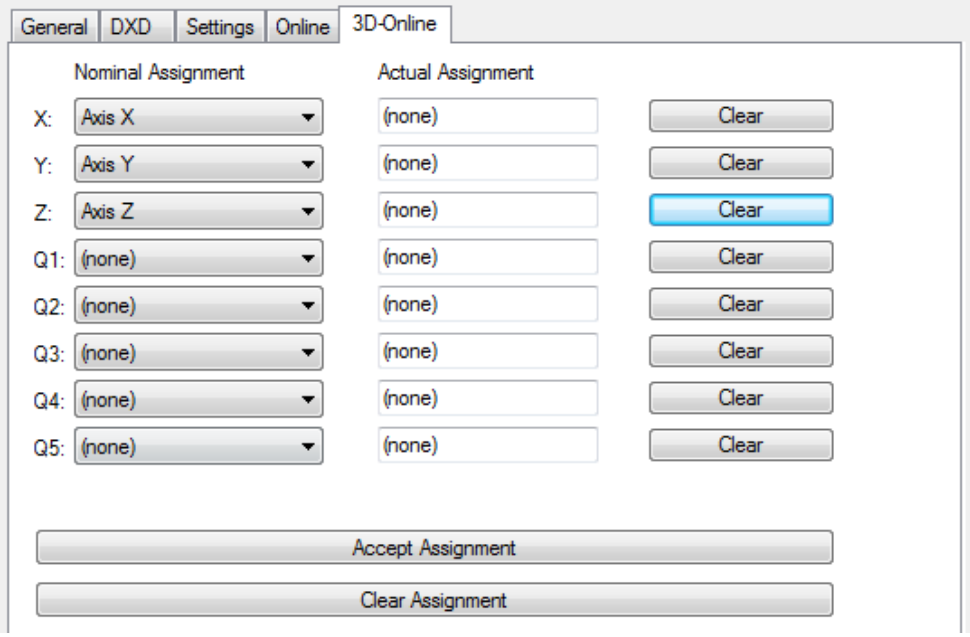


Figure 46 Assigning axed for interpolated movements.

Continuing with the setup, the interpolating system must be able to take control of the axes. This is done on the *Override* tab inside the *Interpolation*'s parameters (Figure 47). There, the user may select between 0 and 100% axis override. A value of 0% means that the interpolating system cannot move the axes, while a value of 100% means that the interpolating system may move the axes at full speed.

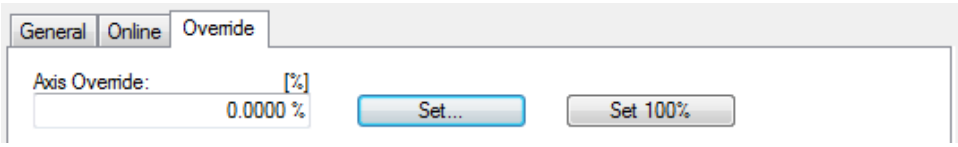


Figure 47 Axis control tolerance for an interpolation system.

By clicking on *GO Interpreter*, the G-code interpreter included in TwinCAT is opened (Figure 48). The parameters window is divided in two: the middle-upper part allows access for configuring the interpreter, while the middle-lower part offers real-time information about the axes (actual position, set position, lag distance, set velocity and error code if the axis is in error state). The most important tabs in the upper part are explained below.

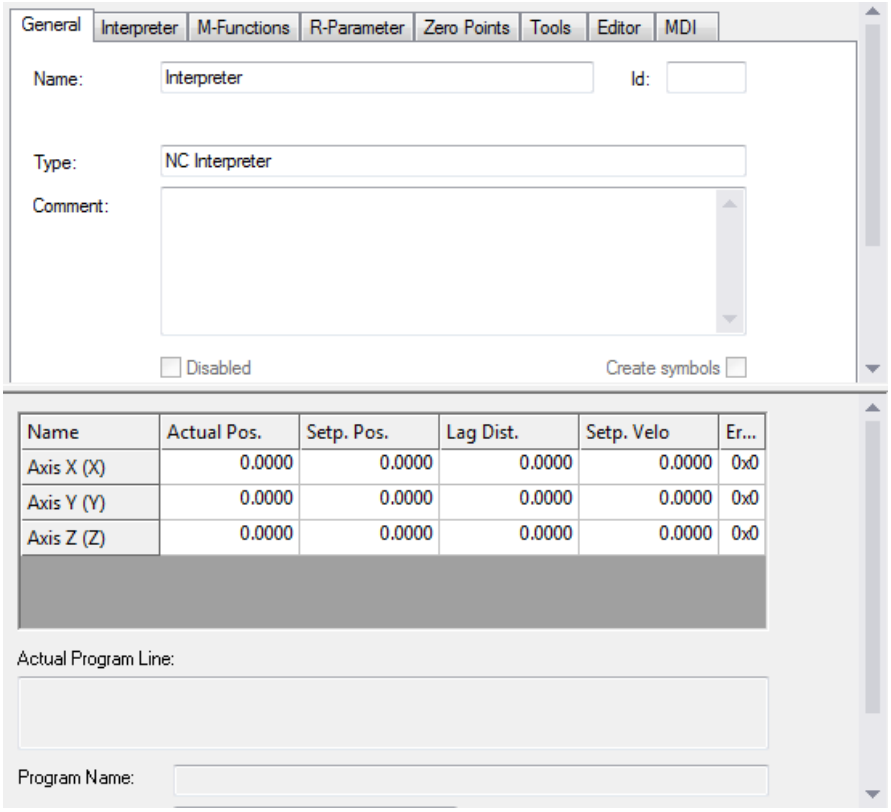


Figure 48 General tab of the GO-Interpreter in TwinCAT.

The *General* tab offers general information about the interpreter, such as its name, type and user comments.

The *Interpreter* tab allows selecting the interpreter type (by default only *NC Interpreter DIN 66025* is installed, although the user may install different interpreters), configure the buffer size and define the G70 and G71 factors (which are used respectively for roughing and finishing pieces).

On the *M-Functions* tab it is possible to configure the machine functions, either by manually entering them one by one or importing the m-functions description file.

The *R-Parameter* tab includes the radius variables, which can be either edited directly on this tab or specified in the G-code file.

On the *Tools* tab it is possible to define the geometry, as well as other characteristics of each of the tools used.

The *MDI* tab allows sending single commands to the interpreter. The user can write the command inside the input text box and press F5 button or Enter. Pressing F6 the system will stop the execution of the current instruction.

By navigating to the *Editor* tab (Figure 49), the user can load and run a ready-made G-code file of type **.nc*. The steps for loading and running a G-code file are: first, the file must be located and this is done by pressing *Browse…* and selecting the file in its containing folder; second, F7 button must be pressed to load the program in the interpreter (the user may check

the program and edit it if necessary, pressing F9 afterwards to save the changes); the last step is to press F5 and the program will start running (Figure 50). F6 will stop the execution of the program and F8 will reset the interpreter (for example, in case of an error).

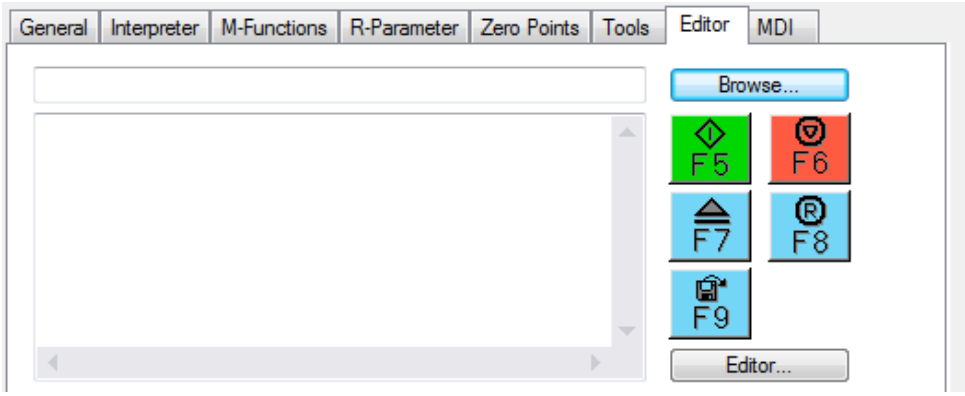


Figure 49 *Editor* tab of the *GO Interpreter*.

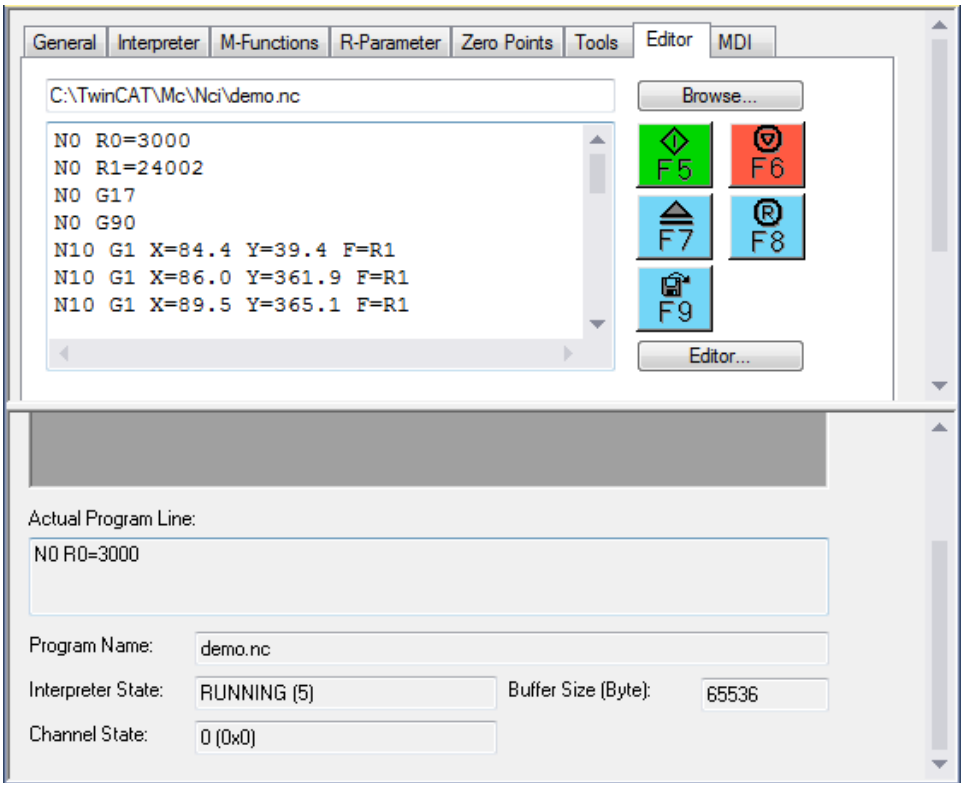


Figure 50 Running G-code.

4 BUILDING A PROTOTYPE

4.1 Design

In order to reduce costs and give a second chance to some materials already present at university, the prototype design had to be redone, although the core idea and configuration was kept as unaltered as possible.

Some of the changes include the use of the *45 Series Profiles* instead of the *60 Series*, different motor and servo drive models, different linear modules for the X and Y-axis, only one linear module for the X-axis and different controller. The tailored made components present in Appendix 3 were adapted to suit the new dimensions of aluminium profiles and overall structure.

4.2 Construction

4.2.1 Metallic frame

The metallic frame was based on a modified version of an already built 2D drawing machine (Figure 51). The first step was to disassemble part of the machine and remove unnecessary parts.

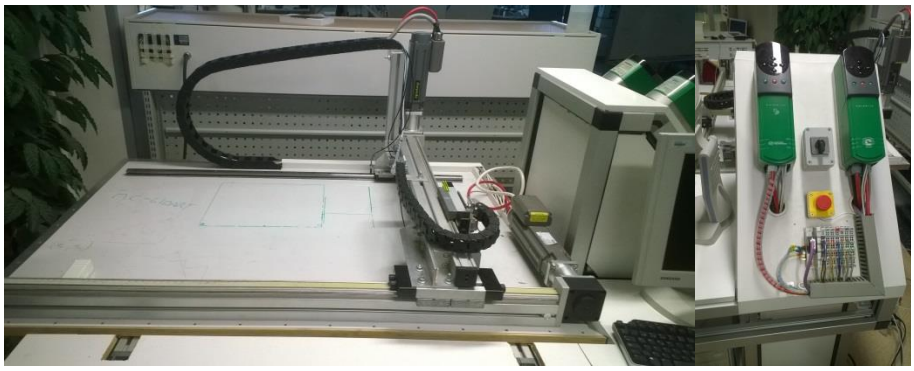


Figure 51 Old 2D drawing machine used for spare parts.

After commissioning the material available for the metallic structure, a 3D model was required in the interest of using them in the most efficient way. Again, the program used was Autodesk Inventor Professional 2015. The reason behind modelling first the structure and only then proceed to cutting and mounting it, was that the 3D modelling software allows for undoing mistakes and preview the end result. Figure 52 presents the metallic structure of the prototype in Inventor.

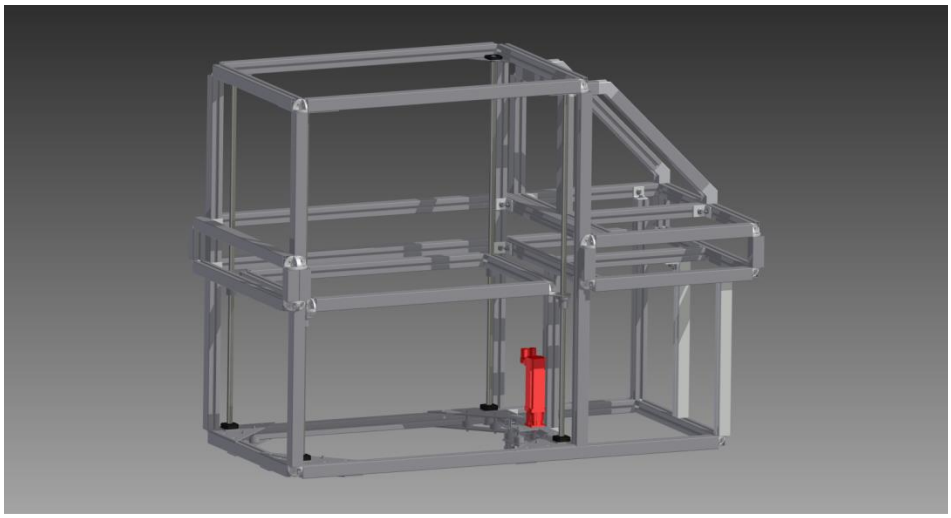


Figure 52 Finished metallic structure of the prototype.

4.2.2 Attaching linear modules and coupling motors

Once the metallic frame was built, the linear modules could be attached, although first they required modifications: the plinth for the motor had to be turned 180º. First, the X-axis linear module was attached on the top most part of the frame using bolts and M8 T-nuts. Next, in parallel at the required distance a linear guide was installed using M4 T-nuts every 20 cm. On top of the carriages from the linear module and the linear guide, the Y-axis linear module was firmly secured using M8 screws, M8 T-nuts and custom made parts.

The Z-axis' ball screws assemblies were securely fixed to the metallic structure using eight triangular plates similar to the ones shown in Appendix 3/1 and 3/2, which in addition provided extra stability to the whole machine.

Next, the motors were attached to their location. For the X and Y-axis, it was sufficient to secure each motor using four M6 screws. For the Z-axis, first the custom made plates for holding the motor were installed. Second, the bottom holding plates for the belt guiding bearings and the T5 belt were installed. In the last step the motor was secured with four screws and the belt tightened. It is important to notice that there are two screws which tighten the belt. This allows correcting a possible vertical drift of the belt by modifying the inclination angle of the tightening bearings. Figure 53 presents the linear modules and motors mounted on the metallic structure.

Figure 53 Metallic frame with the linear modules for the X, Y and Z-axis mounted.

4.2.3 Attaching the platform

Before firmly tightening the screws holding the vertical ball screws, it was necessary to align the axis of the four ball screws. Failing to accurately align the axis would cause additional radial load applied to them, which depending on the magnitude, could have undesirable effects ranging from an increased load on the motor or ball-nuts, to the bending of the screws or blockage of the platform movement.

Once the alignment was achieved, the next step was to mount the Y-shaped parts to the ball-nuts (Appendix 3/6), and finally attach the platform which would hold the working “bed”.

4.2.4 Installing the servo drives and PLC terminals

The servo drives for the X and Y-axis were kept in the same position, while a third servo drive was installed between them. The rail for the PLC and the terminals was increased in length to accommodate the Ethernet bus coupler model BK9000 and the required terminals. For this prototype, the servo drives used are analog driven and therefore additional, specialized terminals were needed. The installed terminals were:

- Three KL5111-0010 terminals to read the encoder values from the motors. The ending “0010” indicates that the terminal accepts A, B and C signals with 5 V logic.
- One KL1408 terminal for the inputs.
- Two KL2134 terminals for the outputs.
- One KL4034 terminal, which includes four channels, ±10 V analog outputs. One output was needed per servo drive.
- One KL9010 terminal, which is the ending terminal and always must be present.

4.3 Programming example

In the NC configuration chapter it was clarified how to set up the system and use G-Code for controlling it. The same code was used to test the prototype as well.

The first step was to configure the axes with the same method used in the previous chapter, except that since a Point-To-Point movement was required, there was no need to configure the interpolation channel. Next, the G-Code program was loaded and executed.

One of the first issues noticed was the need for an improved PID control. This was due to elastic movements present in the structure. Two actions were taken in order to improve the accuracy and control: first, extra reinforcements were added to the structure, and second, the controller was changed to *Position P and velocity PID controller*, which improved significantly the accuracy.

5 CONCLUSION

The main objective for the present thesis project was to design a multipurpose device, capable of working in three dimensions. Industrial quality components were examined and implemented to the design, based on their high quality and reliability.

At the first stage, background research was carried out concerning 3D-printing related topics, such as interpolation and G-code. Afterwards, a suitable control and motion system ought to be found. Once suitable components were found, the machine design could be conducted. By using the Autodesk Inventor design tool it was possible to obtain a 3D model of the device. In addition, the use of such tool favoured a rapid design process since it was faster to implement changes in a virtual model.

Building a prototype proved to be one of the greatest, yet enjoyable challenge experienced by the author. Even though major design changes had to be performed, the resulting prototype could offer essentially the same functions as the original design. For this prototype, a metal engraving tool was used at first for testing purposes, followed by a milling or drilling tool.

All in all, the results met and even surpassed the author's expectations. The design from ground up to completion and even the development of a prototype provided valuable experience and expertise for the author. However, the author recommends that the following improvements be added to the device:

- One of the most important future improvements must be the inclusion of a fail-safe mechanism which could guarantee no harm to an operator or the machine itself. One possibility to achieve this would be to directly attach the Z-axis motor to a linear module and include a sensor to detect when the belt is broken to stop the motors. A second pos-

sibility would be to attach a breaking mechanism to each linear module of the Z-axis and a sensor to activate them and stop the motors.

- The system's automation can be further extended by including an automatic tool exchanging system.
- While configuring the prototype it was noticed that elastic movements of the structure appeared with high dynamic movements. Extra reinforcements in the structure may be desirable.
- The device's functions can be developed with additional tools, such as a 3D printing head or grabbing device.
- Similar to the previous recommendation, added functionality can be achieved via specialized programs and Human Machine Interfaces (HMIs), thus eluding the need of using G-code.

SOURCES

Beckhoff 2012a. Compact servo terminal with One Cable Technology. Verl: Beckhoff Automation GmbH.

Beckhoff 2012b. TwinCAT 3: eXtended Automation (XA). Verl: Beckhoff Automation GmbH.

Beckhoff 2014a. eXtended Automation with TwinCAT 3 Documentation. Verl: Beckhoff Automation GmbH.

Beckhoff 2014b. AM8042 Servomotor 4.1 Nm (standstill torque). Accessed 4 May 2015. http://www.beckhoff.com/english.asp?drive_technology/am8042.htm

Beckhoff 2014c. AM8032 Servomotor 2.38 Nm (standstill torque). Accessed 4 May 2015. http://www.beckhoff.fi/english.asp?drive_technology/am8032.htm

Beckhoff 2015a. Hardware manual CX51x0. Verl: Beckhoff Automation GmbH.

Beckhoff 2015b. CX5100 – Embedded PC series with Intel Atom processor. [online image] Available at: http://www.beckhoff.com/english.asp?embedded_pc/cx5100.htm Accessed 4 May 2015.

Bolton, W. 2009. Programmable Logic Controllers. 5th edition. Oxford: Newnes.

Bosch-Rexroth 2014. How to choose a linear module. Accessed 4 May 2015. http://www.boschrexroth.com/en/us/products/product-groups/linear-motion-technology/linear-motion-systems/linear-motion-systems-overview/how-to-choose-a-module/index

Bosch-Rexroth 2015. Aluminum Framing Shop. Accessed 4 May 2015. http://www13.boschrexroth-us.com/Framing_Shop/Product/Default.aspx?units=1

Bourke, P. 1992. Spheres, equations and terminology. Accessed 16 April 2015. http://paulbourke.net/geometry/circlesphere/

Bourke, P. 1992. Spherical triangles. [online image]. Available at: http://paulbourke.net/geometry/circlesphere/ Accessed 16 April 2015.

Gonen, R. 2013. Stereolithography (3D Printing) Algorithms and Thoughts. [online image]. Available at: https://ravehgonen.wordpress.com/2013/02/19/stereolithography-3d-printing-algorithms-and-thoughts/ Accessed 16 April 2015.

Helms, T. 2011. Typical Brushed Motor in Cross-section. [online image]. Available at: https://experimentalev.files.wordpress.com/2011/03/dc-motor-parts-6.jpg Accessed 19 March 2015.

InfoPLC n.d. TwinCAT NC Configuration. Accessed 4 May 2015. http://www.infoplc.net/files/descargas/beckhoff/infoplc_net_guide_for_twincat_nc_control.pdf

ISO 5725-1:1994. Accuracy (trueness and precision) of measurement methods and results – Part 1. Geneva: International Organisation for Standardization. Accessed 13 Feb 2015. https://www.iso.org/obp/ui/#iso:std:iso:5725:-1:ed-1:v1:en

Kopeliovich, D. (n.d.) Resistance Spot Welding. [online image]. Available at: http://www.substech.com/dokuwiki/doku.php?id=resistance_welding_rw Accessed 15 March 2015.

Marcos M. L. 2010. Technical drawing course. IES Pedro Cerrada 1.1 - 20.12.2010.

Moler, C. 2004. Numerical Computing with MATLAB. Natick, MA: MathWorks.

Rittal 2014. Cabinet model 6901.100. Herborn: Rittal GmbH & Co. KG.

Seward, M. & Zeigler, N. 2012. Handbook of Electric Motors and Electromagnetic Coils. Darya Ganj: Academic Studio.

Sew-Eurodrive 2006. Drive Engineering - Practical Implementation - Servo Technology. Bruchsal: Sew-Eurodrive GmbH & Co KG.

Small Electronic Thingies for All Kinds of Fun Stuff 2010. Resolvers. [online image]. Available at: http://wiki.edwindertien.nl/lib/exe/detail.php?id=modules%3Aresolver&media=modules:02253.png Accessed 19 April 2015.

Tamagawa Seiki CO.,LTD. Absolute Encoder Simplified Structure. [online image] Available at: tamagawa-seiki.com/english/encoder/rotary02.jpg Accessed 19 March 2015.

Tamagawa Seiki CO.,LTD. Incremental Encoder Simplified Structure. [online image] Available at: tamagawa-seiki.com/english/encoder/rotary01.jpg Accessed 19 March 2015.

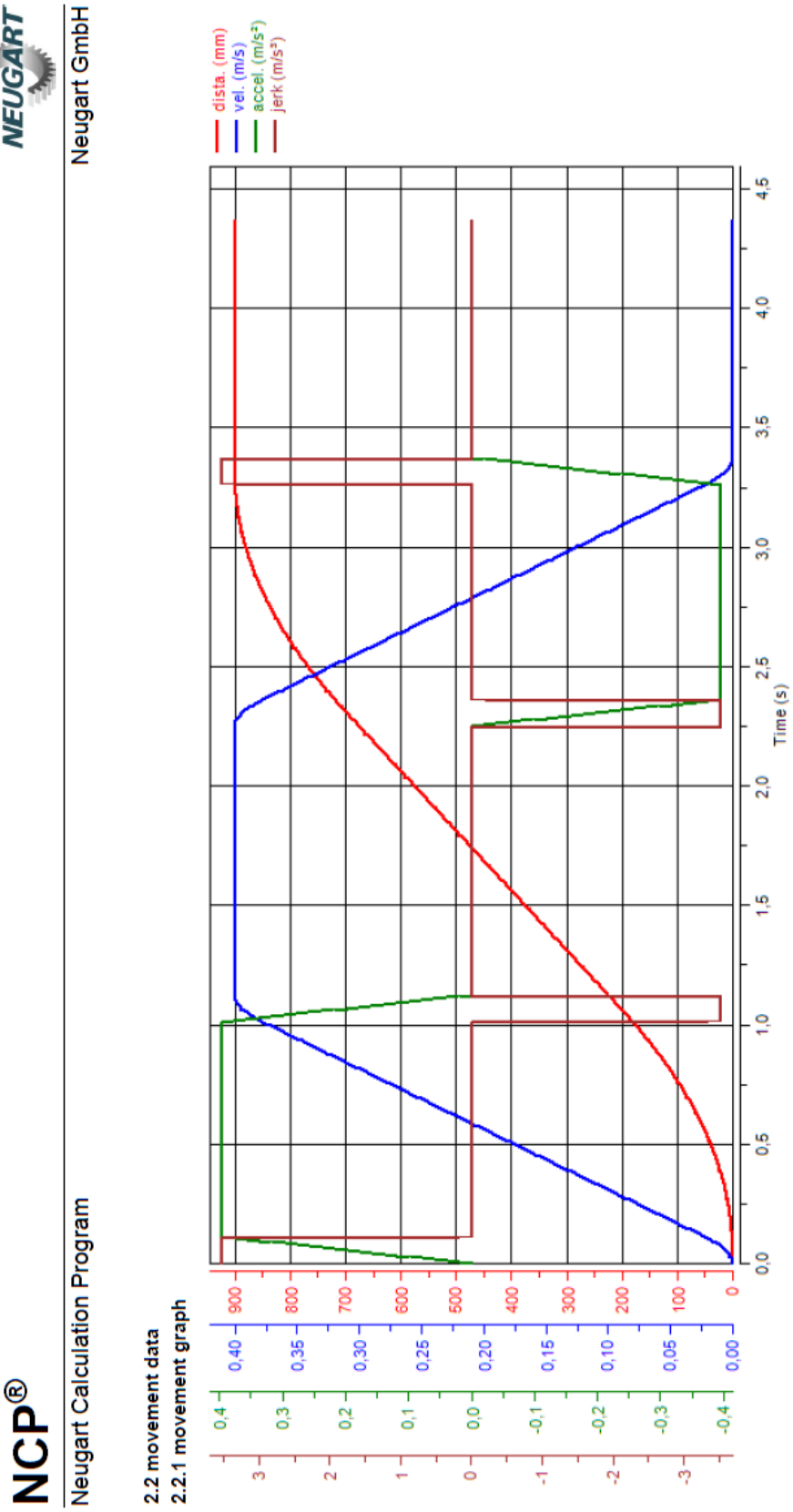
Tierney, H., Murphy, C., Jewell, A., Baber, A., Iski, E., Khodaverdian, H., McGuire, A., Klebanov, N. & Sykes, C. 2011. Experimental Demonstration of a Single-Molecule Electric Motor. Nature Publishing Group.

Universal motor n.d. [online image]. Available at: http://www.transtutors.com/homework-help/electrical-engineering/single-phase-ac-motors/universal-motor.aspx Accessed 19 April 2015.

Uotila, Jukka. Sent 22.1.2015. Re: Project parameters – HAMK. [e-mail message]. Recipient Andrei Sandru. Viewed 27 January 2015.

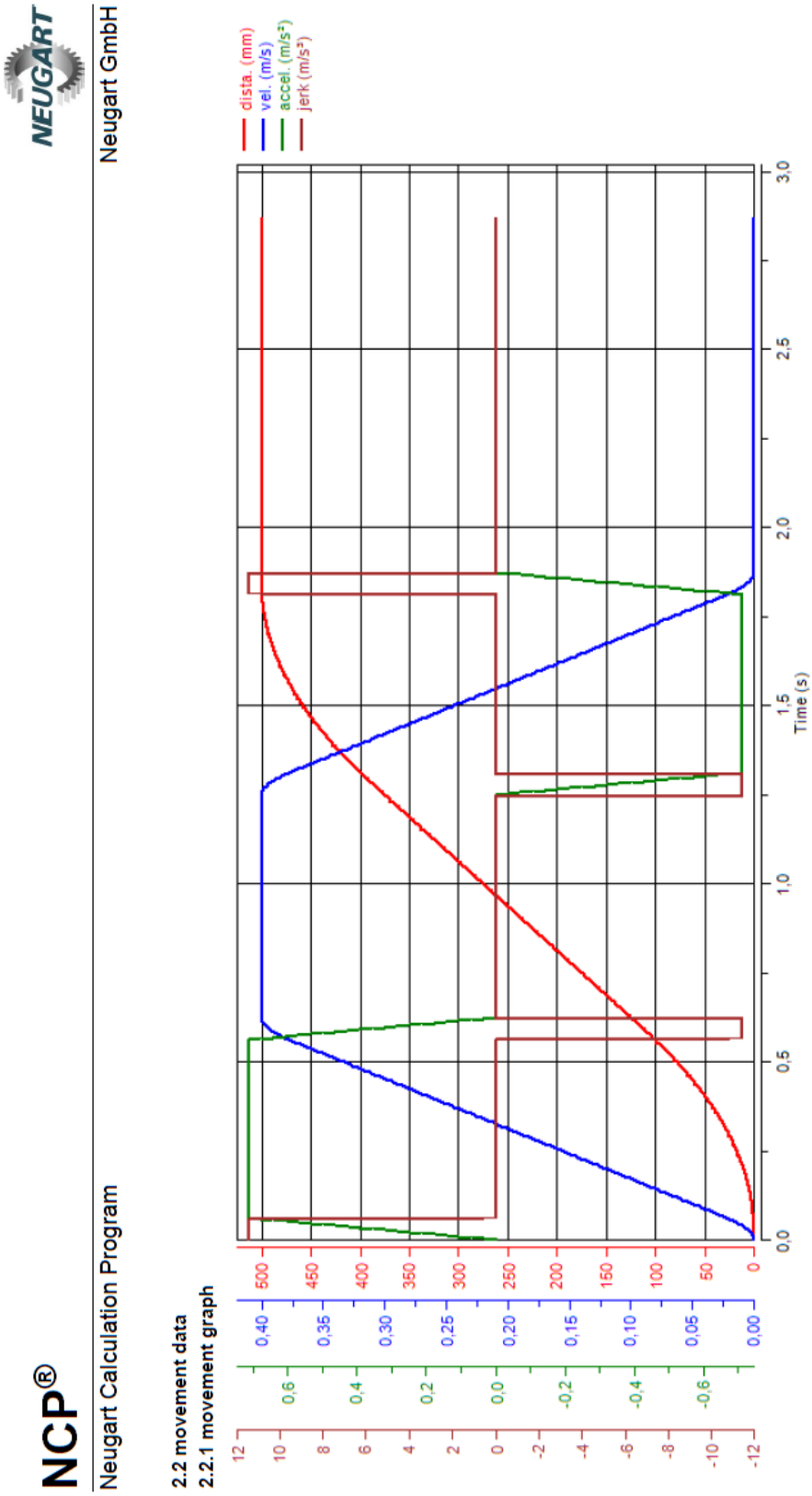
Wahjudi, D. 1999. Precision of movements. [online image]. Available at: http://faculty.petra.ac.id/dwahjudi/private/robot1.htm Accessed 19 April 2015.

DYNAMICS REQUIREMENTS FOR THE X-AXIS



(Uotila 2015.)

DYNAMICS REQUIREMENTS FOR THE Y-AXIS



(Uotila 2015.)

DYNAMICS REQUIREMENTS FOR THE Z-AXIS

(Uotila 2015.)

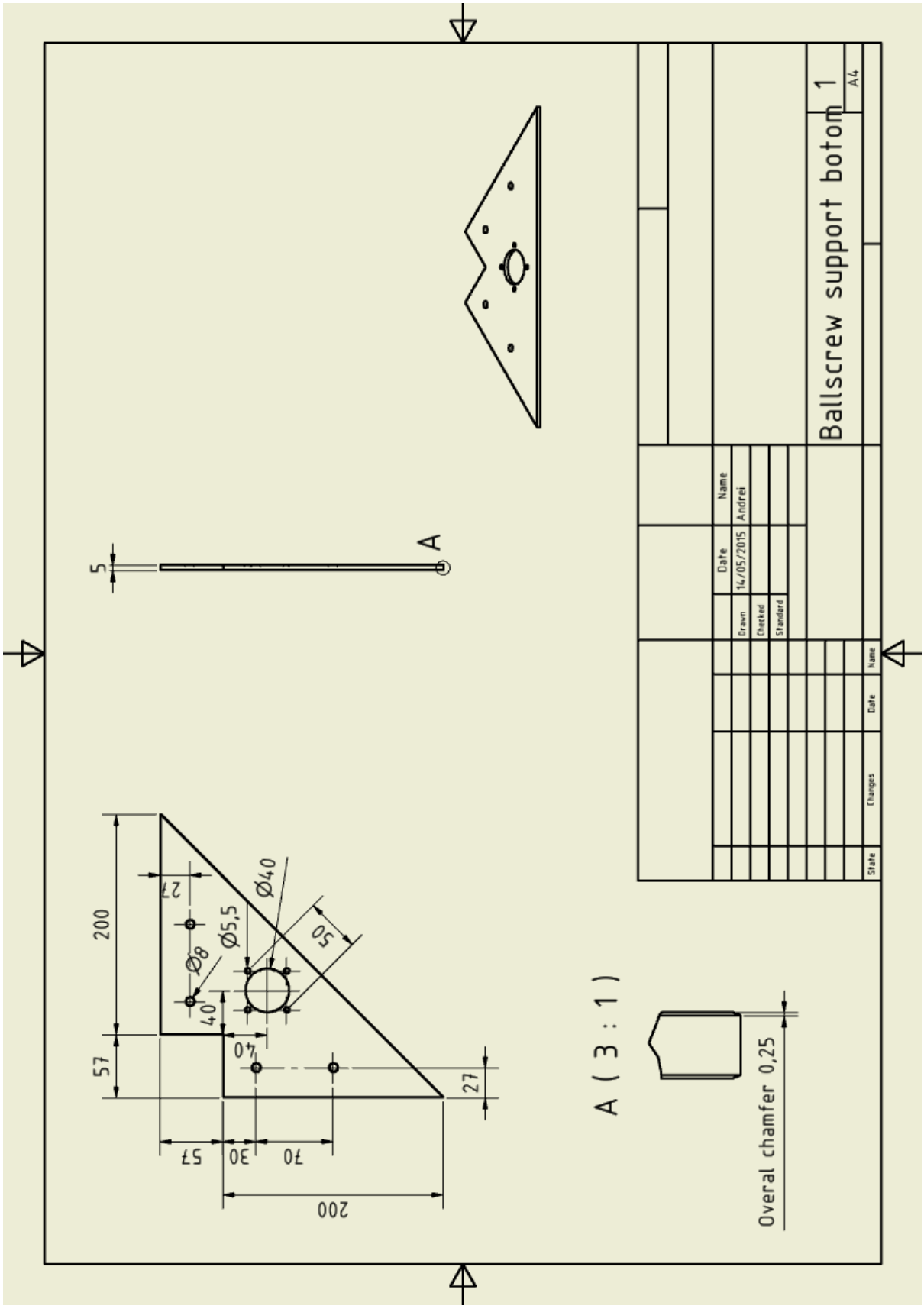
Appendix 2

CX51x0 TECHNICAL DATA

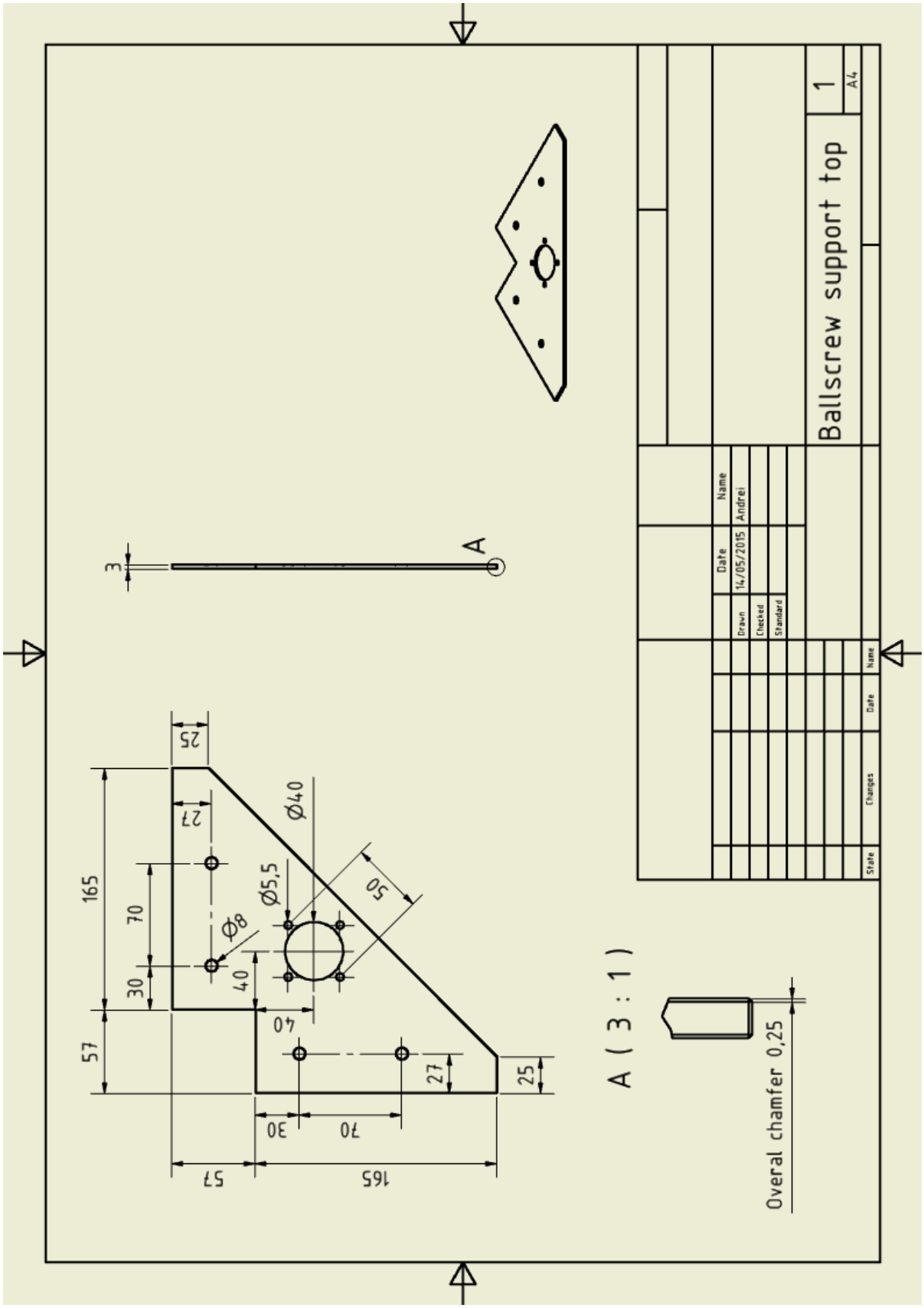
Technical data	CX5120	CX5130	CX5140
Processor	processor Intel® Atom™ E3815, 1.46 GHz, 1 core (TC3: 40)	processor Intel® Atom™ E3827, 1.75 GHz, 2 cores (TC3: 40)	processor Intel® Atom™ E3845, 1.91 GHz, 4 cores (TC3: 50)
Flash memory	slot for CFast card, card not included		
Internal main memory	2 GB DDR3 RAM (not expandable)	4 GB DDR3 RAM (not expandable)	4 GB DDR3 RAM (not expandable)
Persistent memory	integrated 1-second UPS (1 MB on Compact Flash card)		
Interfaces	2 x RJ45, 10/100/1000 Mbit/s, DVI-I, 4 x USB 2.0, 1 x optional interface		
Diagnostics LED	1 x power, 1 x TC status, 1 x flash access, 2 x bus status		
Clock	internal battery-backed clock for time and date (battery exchangeable)		
Operating system	Microsoft Windows Embedded Standard 7 P		
Control software	TwinCAT 2 PLC runtime or TwinCAT 2 NC PTP runtime		
Power supply	24 V DC (-15 %/+20 %)		
Dielectric strength	500 V (supply/internal electronics)		
I/O connection	E-bus (EtherCAT Terminals) or K-bus (Bus Terminals), automatic recognition		
Current supply E-bus/K-bus	2 A		
Max. power loss	9 W (including the system interfaces)	11 W (including the system interfaces)	12 W (including the system interfaces)
Dimensions (W x H x D)	124 mm x 100 mm x 92 mm	142 mm x 100 mm x 92 mm	142 mm x 100 mm x 92 mm
Weight	approx. 860 g	approx. 960 g	approx. 960 g
Operating/storage temperature	-25…+60 °C/-40…+85 °C		
Relative humidity	95 %, no condensation		
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27		
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4		
Protection class	IP 20		
Approvals	CE		
TC3 performance class	performance (40); please see here for an overview of all the TwinCAT 3 performance classes	performance (40); please see here for an overview of all the TwinCAT 3 performance classes	performance plus (50); please see here for an overview of all the TwinCAT 3 performance classes

(Beckhoff 2015a)

TAILORED MADE PARTS

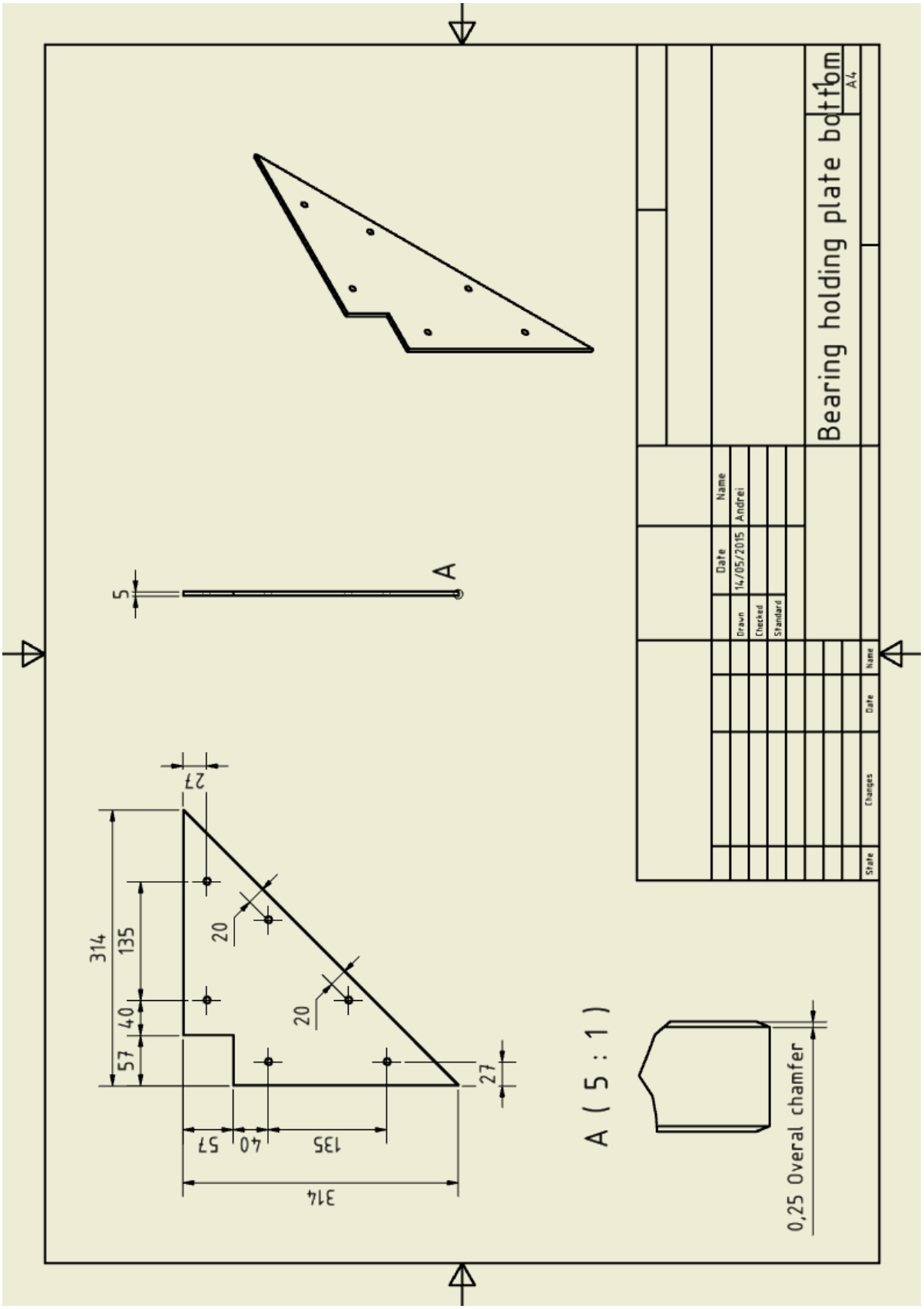


TAILORED MADE PARTS

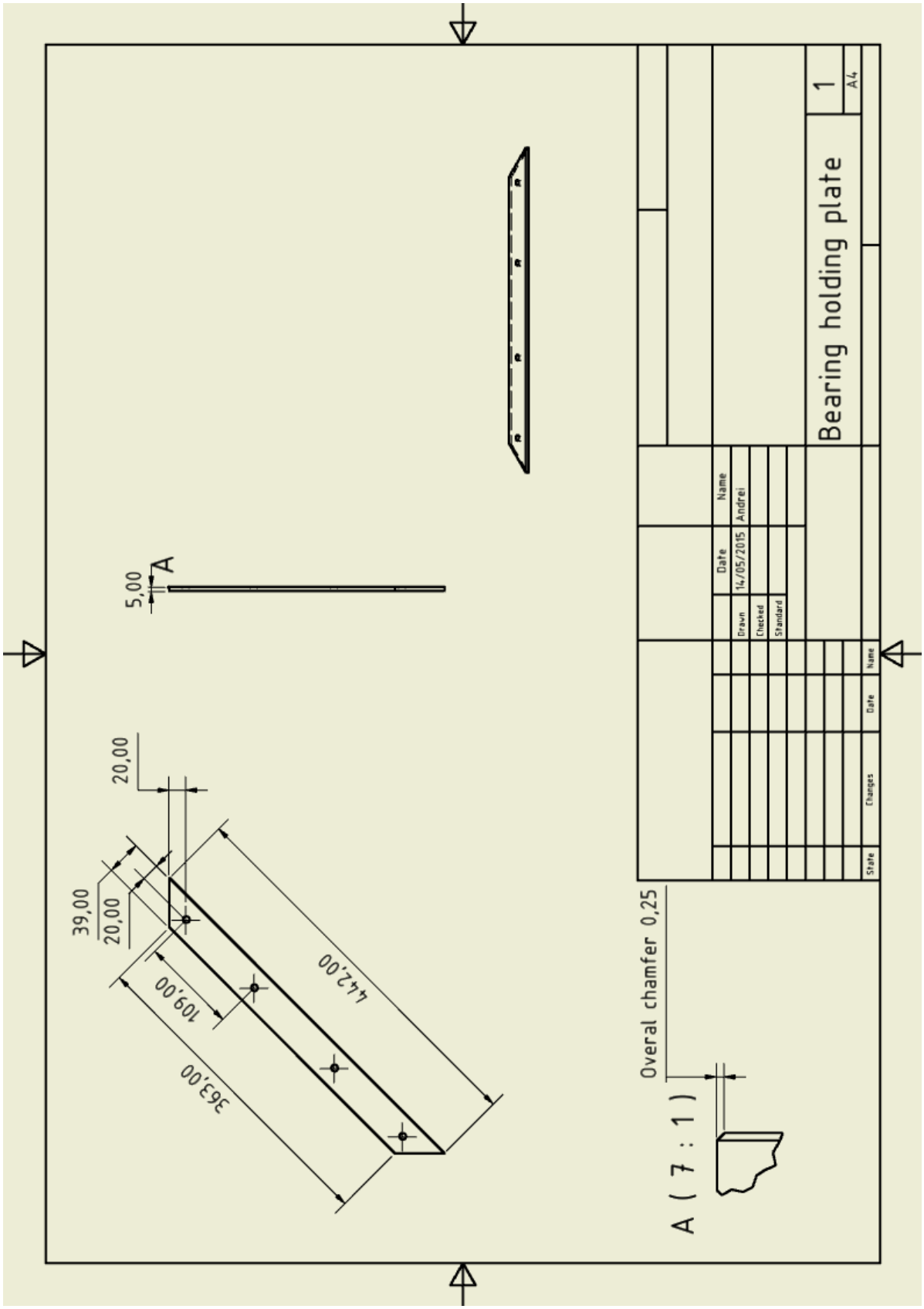


Appendix 3/3

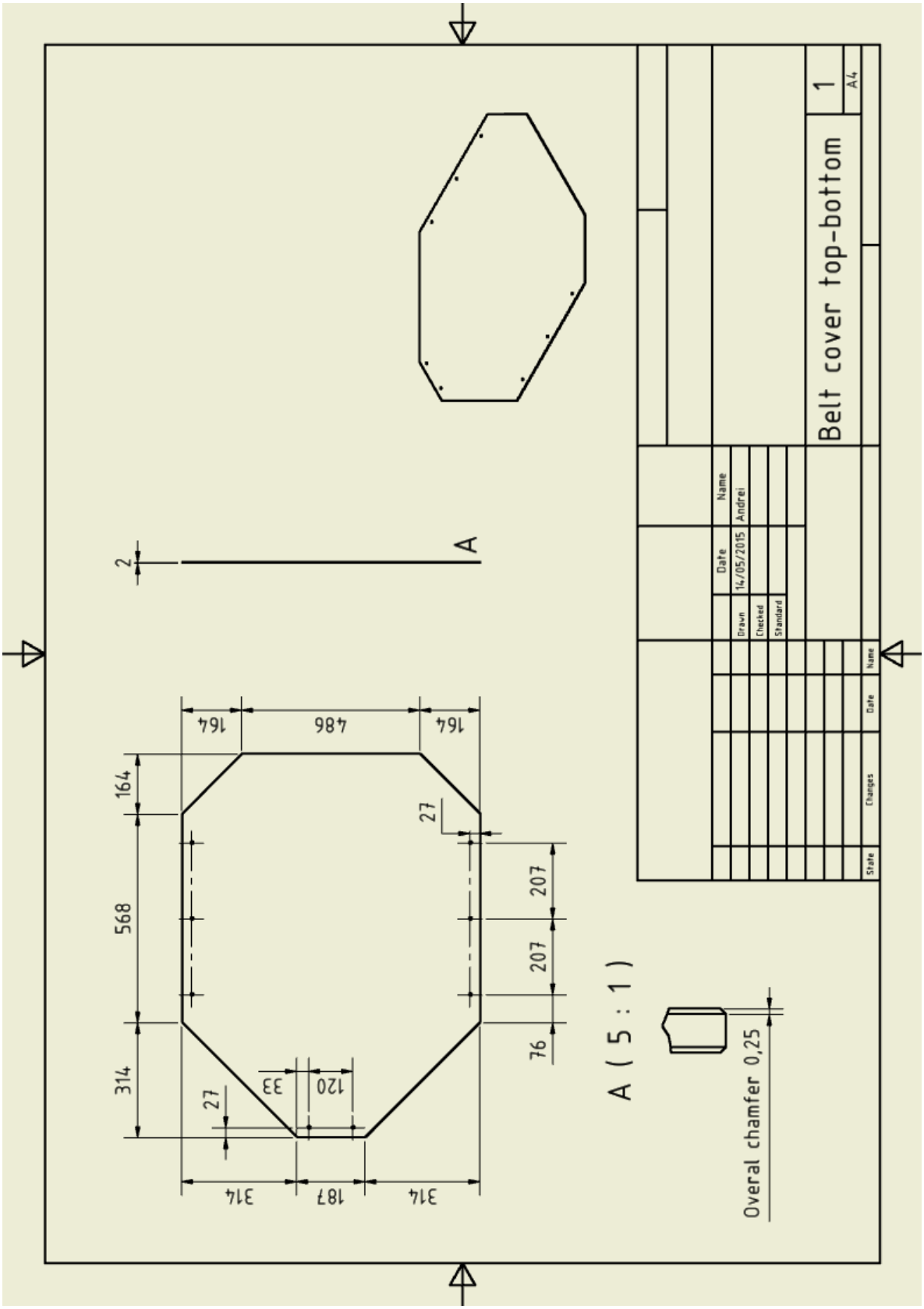
TAILORED MADE PARTS



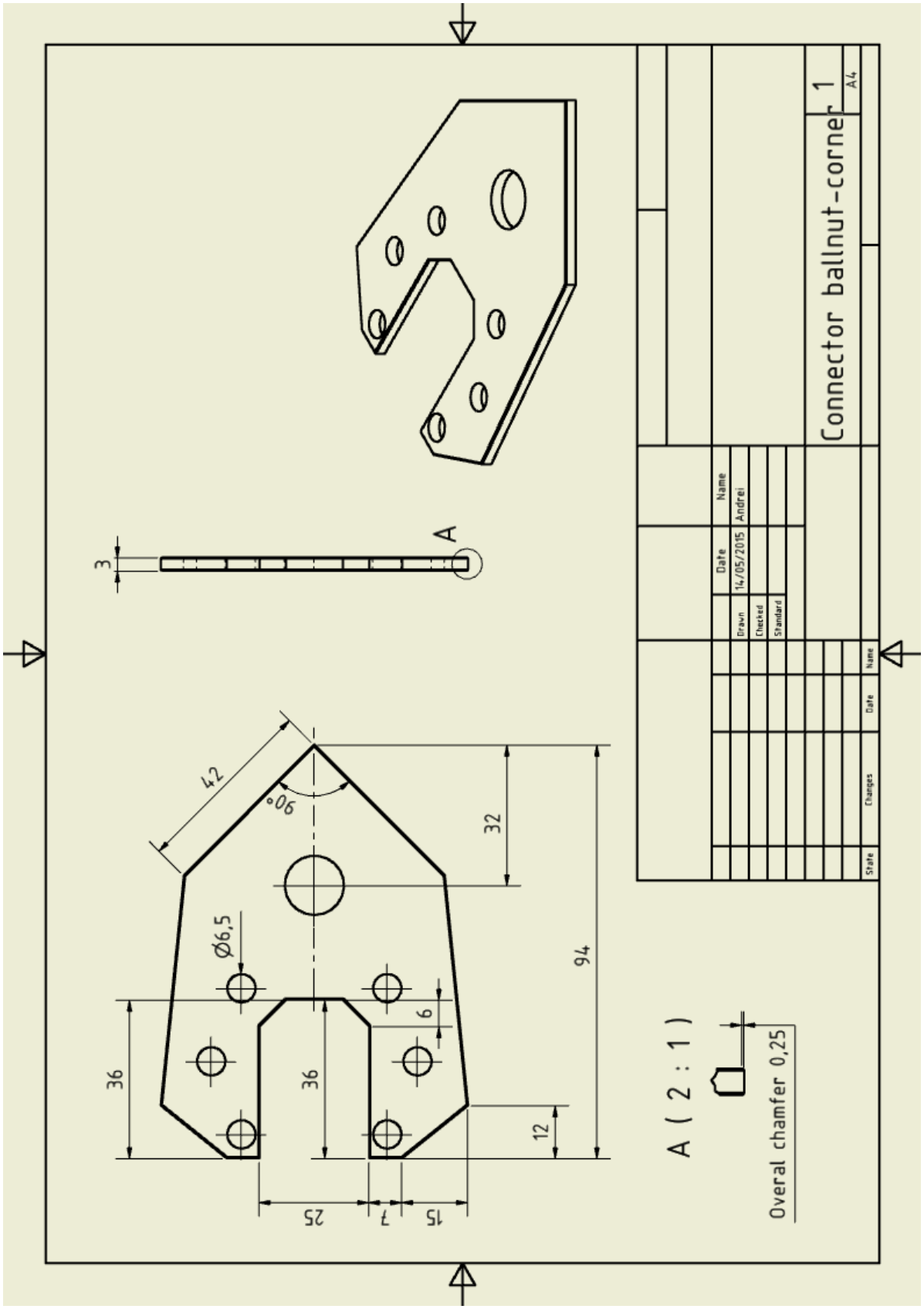
TAILORED MADE PARTS



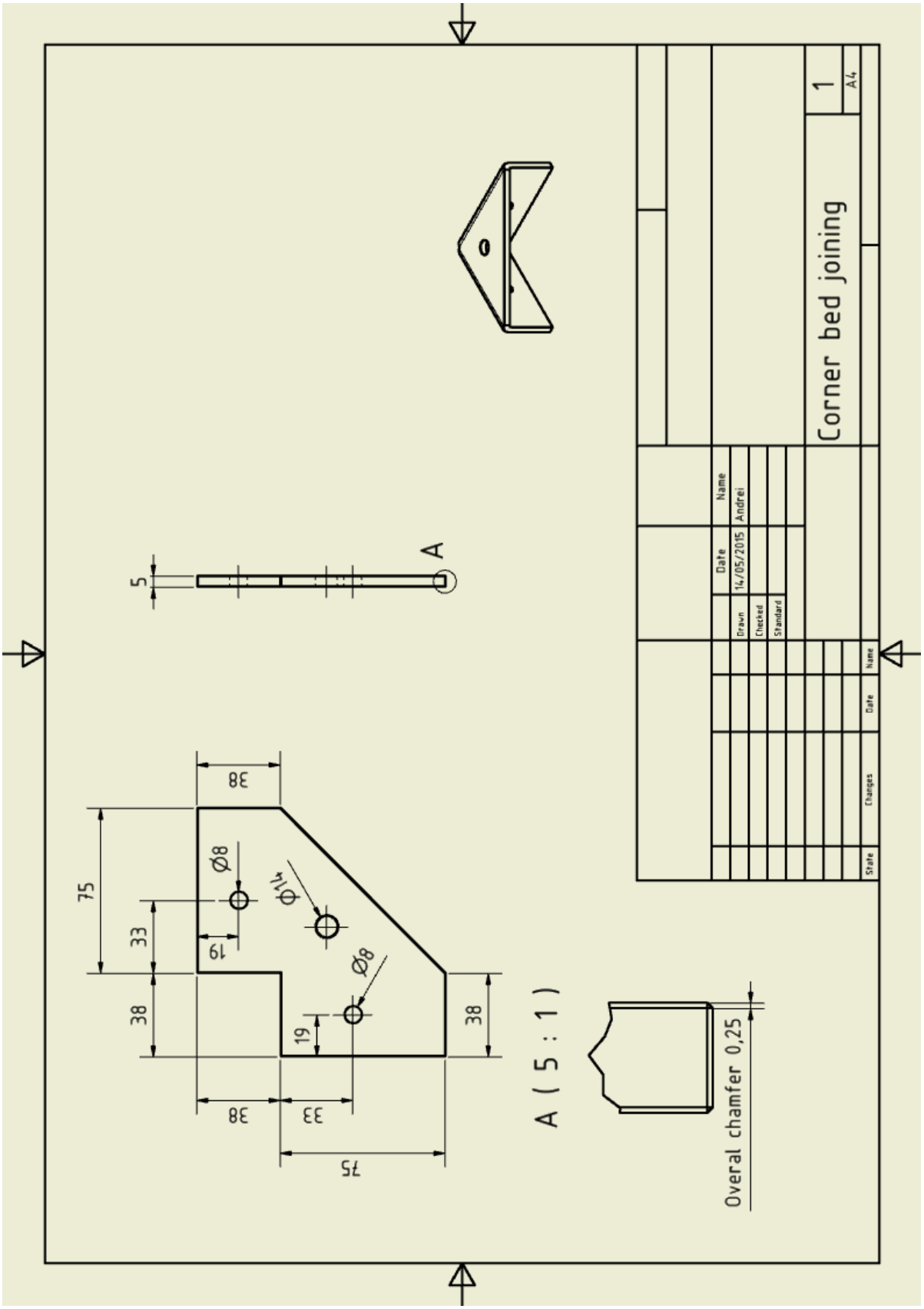
TAILORED MADE PARTS



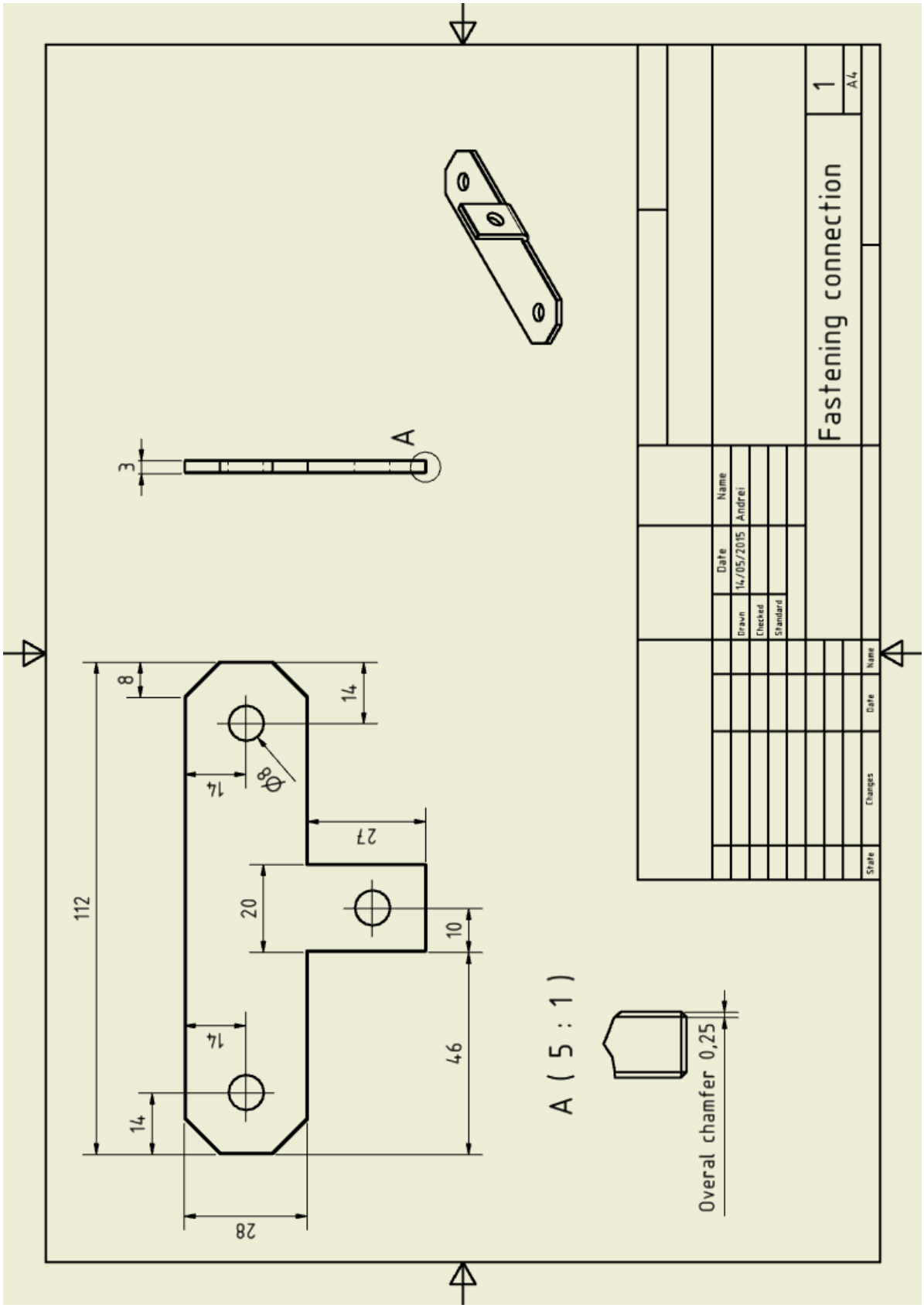
TAILORED MADE PARTS



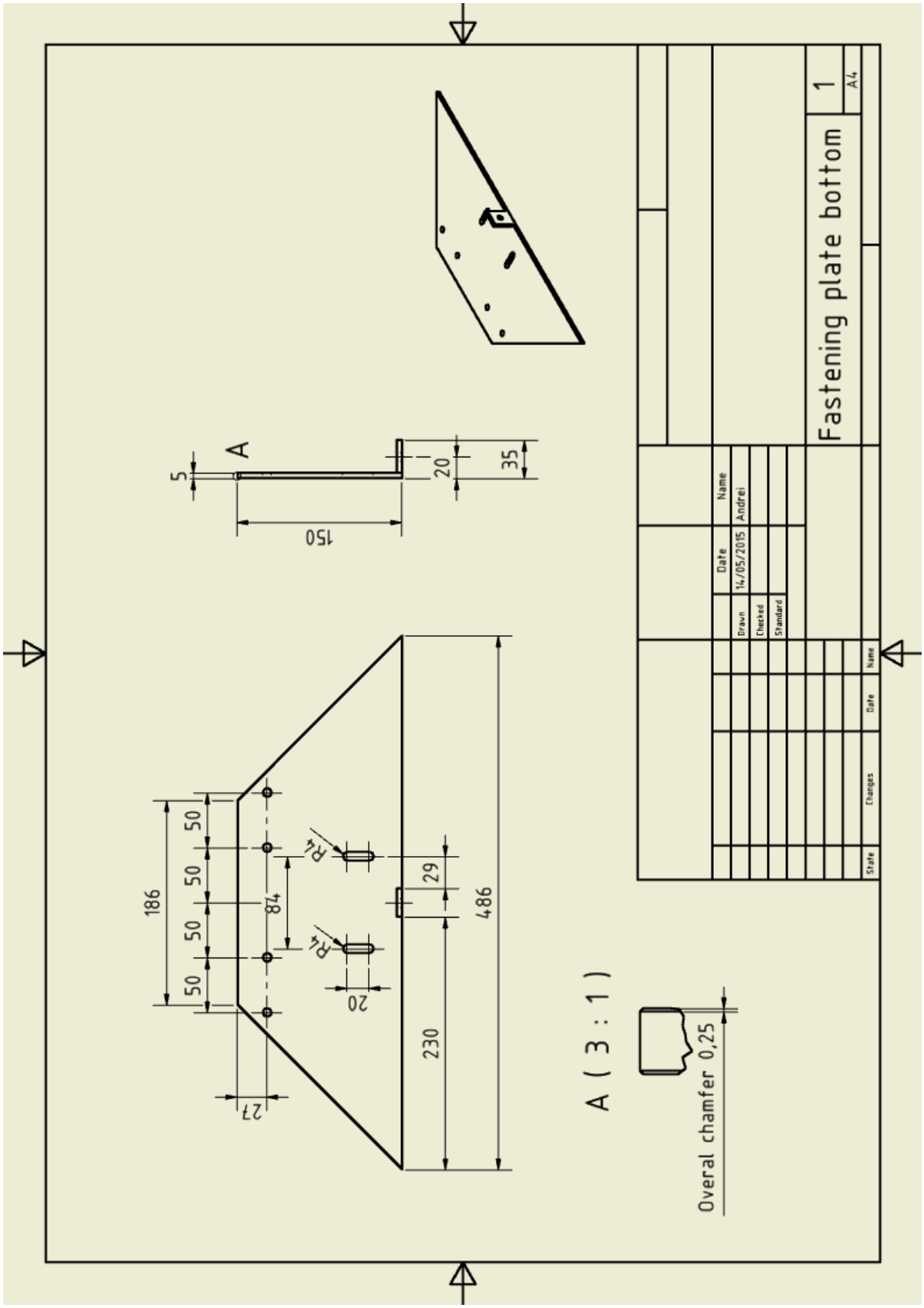
TAILORED MADE PARTS



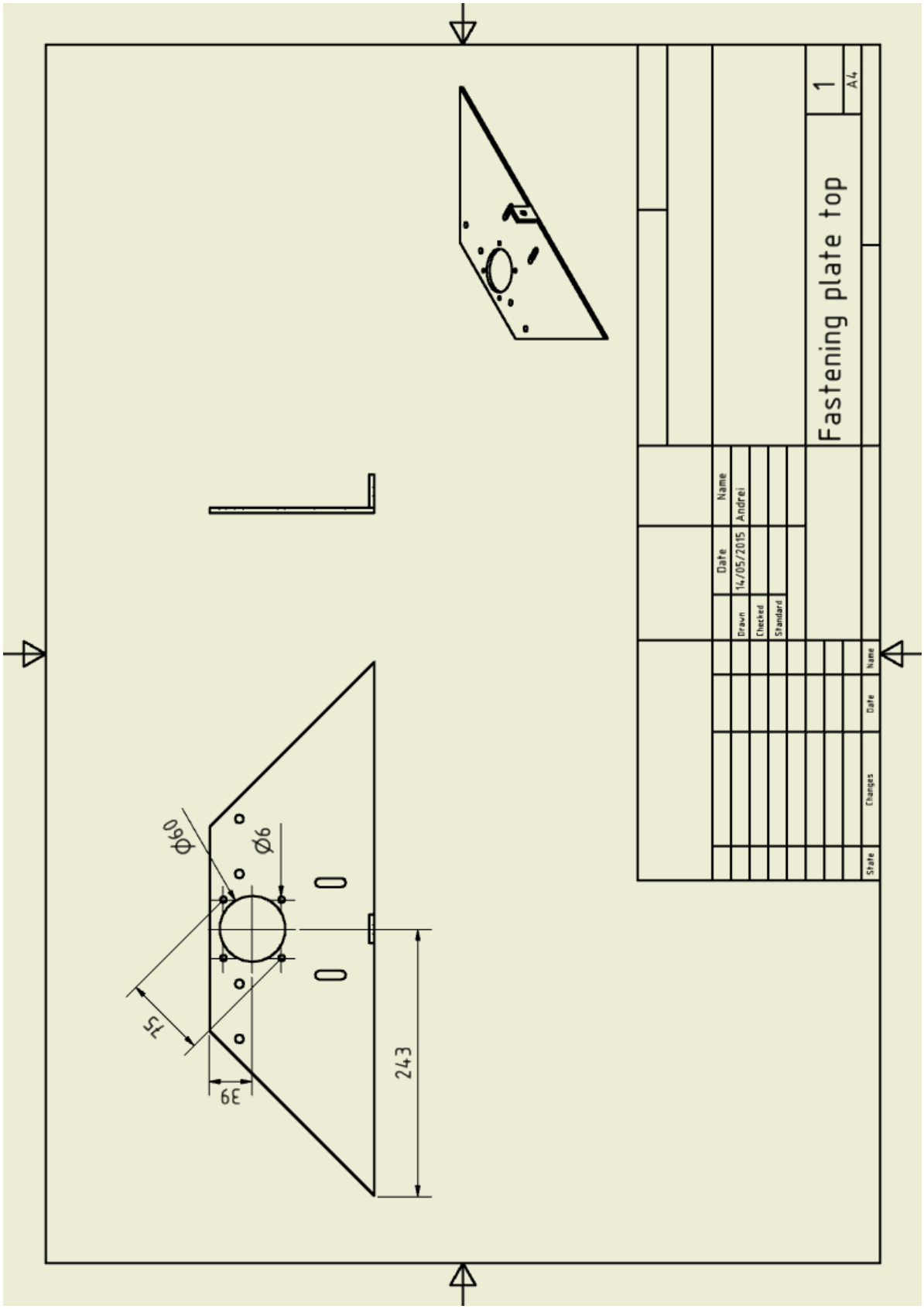
TAILORED MADE PARTS



TAILORED MADE PARTS



TAILORED MADE PARTS



Appendix 4

TECHNICAL DRAWING OF COMPONENTS INSIDE THE CABINET