LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
School of Energy Systems
Electrical Engineering

Jaakko Roima

DEVELOPMENT OF A MOTION CONTROL SYSTEM FOR PROCESSING WITH ULTRA-HIGH SPEED LASER

Examiners: Post Doctoral Researcher, Hamid Roozbahani
Professor, Juha Pyrhönen

ABSTRACT

Lappeenranta University of Technology
School of Energy Systems
Electrical Engineering

Jaakko Roima
Development of a motion control system for processing with ultra-high speed laser
Master's Thesis
2016

Examiners: Post Doctoral Researcher, Hamid Roozbahani
Professor, Juha Pyrhönen

69 pages, 47 figures, 5 charts and 2 appendices.

Keywords: laser processing; laser; linear motor; automation; motion control; servo motor; LabVIEW; SoftMotion; solar cell; CIGS

The main goal of this thesis work was to develop a control system for a laser processing system to perform various scribing and marking tasks. This system includes the motion control of a multi-axial platform for the workpiece and a separate vertical axis for the laser scan head, the control of the scan head to perform scribing tasks and the control and communication with the laser source to set the various parameters. The program was implemented in National Instruments LabVIEW development environment. Before the development of the program, a suitable multi-axial linear motor system for the movement of the workpiece and a servo motor for the vertical movement of the scan head had to be researched and acquired.

First, the interfaces to communicate to the different devices were implemented. The servo drives are communicated via EtherCAT fieldbus, the laser via serial port and the scan head with a special RTC4 card. Then, the program was developed so that the system is automated and the different parts work together in synchronization to perform the desired scribing tasks, such as CIGS solar cells scribing.

In the end, a program was developed which combines the communication of all of the components into one program, increases the automation level of the system and makes all the components work in synchronization with each other to allow scribing and marking.

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Energy Systems
Sähkötekniikan koulutusohjelma

Jaakko Roima
Development of a motion control system for processing with ultra-high speed laser
Diplomityö
2016

Tarkastajat: Tutkijatohtori Hamid Roozbahani
Professori Juha Pyrhönen

69 sivua, 47 kuvaa, 5 taulukkoa ja 2 liitettä.

Hakusanat: lasertyöstö; laser; lineaarimoottori; automaatio; liikkeenohjaus; servomoottori; LabVIEW; SoftMotion; aurinkokenno; CIGS

Tämän työn tavoitteena oli lasertyöstöjärjestelmän automatisointi sekä liikkeenohjausjärjestelmän kehittäminen. Ohjelman kehitykseen kuului käyttöliittymien ja tiedonsiirtorajapintojen ohjelmointi järjestelmän eri osille sekä niiden yhteistoiminnan mahdollistaminen niin että ohjelmalla voidaan tehdä laserkaiverruksia käyttäen lineaari- ja servomoottoreita sekä laserskanneria. Ohjelma toteutettiin National Instrumentsin LabVIEW -kehitysympäristössä. Osana työtä ennen ohjelman kehittämistä oli myös työkappaleen liikuttamiseen soveltuvan lineaarimoottorijärjestelmän sekä vertikaaliselle liikkeelle soveltuvan servomoottorin selvittäminen sekä asentaminen.

Ensin ohjelmaan kehitettiin tiedonsiirto sekä ohjaus kaikille järjestelmän eri osille. Servokäyttöihin kommunikointi tapahtuu EtherCAT -kenttäväylän kautta, laserille sarjaportin kautta sekä laserskanneriin RTC4 -sovitinkortin kautta. Tämän jälkeen ohjelmaan kehitettiin automaatiota niin, että eri osat toimivat synkronoidusti yhdessä mahdollistaen eri tyyppisten laserkaiverrusten tekemisen muun muassa CIGS-aurinkokennoja varten.

Työn lopputuloksena saatiin ohjelma, joka yhdistää kaikki järjestelmän osat toisiinsa ja mahdollistaa automatisoitujen laserkaiverrusten toteuttamisen järjestelmän avulla sekä kasvattaa käytössä olevaa työskentelyaluetta laserskannerin alueesta lineaarimoottorin liikealueen kokoiseksi.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

CIGS	Copper-Indium-Gallium-Selenide
DC	direct current
DLL	dynamic link library
EtherCAT	Ethernet for Control Automation Technology
IEC	International Electrotechnical Commission
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LBM	linear brushless motor
LIM	linear induction motor
LSM	linear synchronous motor
MAC	Ethernet Media Access Controller
PDO	Process Data Object
PM	permanent magnet
PV	photovoltaic
TCO	transparent conducting oxide
TFSC	thin-film solar cell
VISA	Virtual Instrument Software Architecture

Symbols

v	speed
v_{s}	synchronous speed
f	frequency
ω	angular velocity
τ	pole pitch
F_{x}	thrust
P_{out}	output power

1 INTRODUCTION

Laser processing systems are widely used for example in removal of material, sealing of components and surface improving of materials. The laser processing system developed in this thesis is used for scribing and marking. The target is mostly in scribing of Copper-Indium-Gallium-Selenide (CIGS) solar cells but also other materials and methods such as color marking of metals are researched in the future. Generally in laser processing systems, some kind of motion system is usually employed to move the workpiece, scan head or both to achieve desired scribes, handle larger workpieces and working areas and to automate production and make it faster.

The first goal of this thesis work is to research the market and acquire suitable motion systems for the linear two-axis movement of the workpiece and for the vertical movement of the scan head. Linear motors are selected for moving the workpiece as they are best suited for this application because of their accuracy, repeatability, quick movement and good dynamics compared to other linear actuation methods.

After the installation of the motion systems, a program needs to be developed and programmed to control the motion systems and other parts of the laser processing system together and automate the system so that the desired scribes and other necessary functionality are possible to accomplish with the program. LabVIEW development environment is used for the programming since it makes it easy to communicate with many different protocols and libraries at the same time. Creating user interfaces is also straightforward in the environment.

The processing system can also currently be used without any kind of motion system and only with the scan head, but in that case the functionality and automation level of the processing is very limited. In addition, to adjust the focus point of the laser beam, it is useful to be able to position the scan head vertically for different kind of workpieces.

In the end of this thesis, a program was developed that combines the communication of all of the components of the laser processing system into one program, increases the automation level of the system and makes all the components work in synchronization with each other to allow scribing and marking. In addition, some research was conducted on solar cell scribing and color marking, and the software was developed with the future research and testing in mind so that it may be continued in the future.

1.1 Linear motors

Linear motors are basic tools in horizontal position-control systems. They will be used also in the solution proposed in this work. Linear motor is a type of electric motor which generates a linear force along its length. This is in contrast to a rotating electric motor which produces torque. A linear motor can simply be thought of as a rotating electric motor that has had its stator and rotor straightened. The unrolled rotor i.e. the mover moves in a linear path along the stator instead of rotating. Linear motors are an alternative to rotating electric motors in applications where linear movement is needed, such as in the laser processing system developed in this thesis.

1.1.1 Comparison of linear actuation methods

Traditionally without linear motor technology the common ways to produce linear motion from rotary motion have been a lead screw, belt and pulley, ball screw and rack and pinion systems.

The belt and pulley system has limited thrust due to the belt having a certain tensile strength. Moreover, the system is somewhat inaccurate due to the mechanical windup, gearbox backlash and belt stretching. Settling time is also quite poor due to the extensibility of the belt.

Rack and pinion systems are usually more mechanically rigid compared to belt and pulley systems. Nevertheless, the lack of accuracy and repeatability can be a problem depending on the application. The backlash also prevents the encoder system located on the motor from detecting the load position accurately.

Lead screw system is an inexpensive but also inefficient way to generate linear motion. The low efficiency is caused by the friction of nuts riding the screw. This makes it a poor choice for high duty cycle applications as the nuts and screw wear. Repeatability and accuracy are again poor since the parts are not precision-made and the friction causes heat reducing accuracy even more.

On the other hand, ball screw system employes a ball nut on the screw. They are significantly more efficient, even up to 90 percent of the input. Accuracy and repeatability can be improved by precision-ground balls. Even then, they have wear over time which slowly reduces the system accuracy. Either one of the screw systems cannot achieve high speeds while maintaining good resolution. (Hyatt, n.d.)

The advantages of using a linear motor in a linear application are that the motor directly generates linear force and there is no need for any conversion steps. Because of this, there are less moving parts, usually zero backlash and less wearing of components. In addition, the inertia of the system is smaller which improves the control dynamics. The achievable accuracy and repeatability is much higher than with rotating drives, up to micrometer level depending on the encoder type and setup. Table 1 compares the properties of a linear application implemented with traditional methods compared to a linear motor. It can be seen that the linear motor outperforms traditional methods in every aspect except the price. Linear motors can be easily scaled to fit different applications, and very high accelerations and speeds are possible due to the dynamics of the system.

Table 1. Comparison of the properties of linear motor to other methods of producing linear motion. (Ponomarev, 2009)

	Belt and pulley	**Rack and pinion**	**Lead screw**	**Ball screw**	**Linear motor**
Accuracy	--	-	++	+	++
Speed range	+	-	--	-	++
Travel range	-	+	--	--	++
Thrust	-	+	-	+	++
Friction	+	+	--	+	++
Maintenance	+	+	-	--	++
Life time	+	-	--	+	++
Price	+	++	++	+	-
Efficiency	+	-	--	-	++

The biggest disadvantage of linear motors is the price. The biggest factors increasing the price are the high price of magnetic material needed for constructing the motor and the difficulties in manufacturing the structure and mechanics. The efficiency of linear motors is also quite low so a proper cooling system may have to be set up for the coils. Another issue that has to be taken into account is the safety. Linear motor system is usually almost frictionless, so there has to be some way to stop the motor in case of power loss to prevent injuries and damage to the equipment. Previously, the price has been a limiting factor for the expansion of the market, but nowadays the linear motor solutions are getting more cost effective and the need for such high accuracy applications is increasing.

1.1.2 Structure of a linear motor

An electrical motor consists of a stator and a rotor. Stator refers to the stationary part while rotor is the moving part of the system. The part that generates the traveling magnetic field can be referred as the forcer or armature. In addition, when referring to linear motors, we can use the concepts primary and secondary. In this case the primary part refers to the part where the coils that generate the movement are located while secondary refers to the part generating the magnetization. These two terms do not identify which part is actually moving. The rotor can be either the primary part or the secondary part depending on the construction.

There are many variables and choices regarding the linear motor construction and selection. The appropriate selection for a specific application depends on the required specifications and the operating environment of the system. We can divide the motors into different categories for example based on the mechanical structure and shape of the motor or the arrangement and type of the magnets and coils used. Linear motors may be shaped U-channel, tubular or flat.

U-channel shaped motor has two parallel magnet tracks surrounding the mover between the tracks. The mover is ironless and has a low mass which allows for high acceleration. The coil winding is usually three-phase and with brushless commutation. This means that this linear motor is best suited for low stroke distance where high acceleration is needed. The magnet track has minimal magnetic flux leakage as the magnets are facing each other and housed in the U-channel. The length of the magnet track can be easily increased with the only limiting factors being the cable management system, encoders and the difficulties in machining the motor.

In tubular motors, the cylindrical forcer slides along a bar inside of which the magnetization is located. Tubular motors where the first successful systems to find real commercial applications. The construction is similar to a moving magnet actuator with the difference that the coils are replicated which increases the stroke. This construction can’t be used in applications which are sensitive to magnetic flux leakage.

Flat linear motors are brushless and they can be divided into slotless iron, slotless ironless and slotted iron motors.

Slotless ironless motor (or air core motor) has an aluminum base to which coils have been attached. As there is no iron in the rotor, these type of motors have no cogging or attractive forces, which improve bearing life. These motors are best suited for smooth velocity control but output the lowest force of all flat linear motor types. Flat tracks also have high magnetic flux leakage and care must be taken to avoid injury and damaging of the motor or other equipment from the magnets.

Slotless iron motor construction is equivalent to the air core motor with the exception that in addition of the aluminum base the coils are also mounted to iron laminations. The magnetic field can be directed using the iron which increases output force. The negative side of this is that because of the iron, an attractive force is present between the track and the rotor. In other words, there is a cogging force present in the motor unlike in the ironless version. However, the advantage is that the motor produces more force due to the iron used.

To create the slotted iron motor structure, the coil windings are put inside a steel structure. Again, the iron used adds the force output greatly. The disadvantage of this is that an attractive force is present, however it can also be used for benefit as a preload for an air-bearing system. Also cogging forces are present due to the iron. (Aerotech Inc, 2010)

1.1.3 Operating principle

Linear motor transforms electrical energy into mechanical energy. Like rotating electrical machines, linear motors are either linear induction motors (LIM) or linear synchronous motors (LSM). In this chapter the focus is on the working principles of linear synchronous motors.

In a linear synchronous motor, the physical motion is synchronized with the magnetic field. In other words, the actual speed of the motor is equivalent with the speed of the traveling magnetic field. The thrust is produced by the traveling magnetic field generated by a polyphase winding or the magnetic field produced by electronically switched DC windings and their interaction with an array of magnetic poles.

Permanent magnet (PM) brushless LSMs are the most commonly used motors. They can be divided into motors where the input currents are sinusoidal and produce a traveling magnetic field and linear brushless motors (LBMs) where input currents are either rectangular or trapezoidal and exactly synchronized with the position and speed of the rotor.

When the linear synchronous motor operates on the traveling magnetic field principle, the speed v of the rotor

$$v = v_{\mathrm{s}} = 2f\tau = \frac{\omega}{\pi}\tau \tag{1}$$

is equal to the synchronous speed v_{s} of the traveling magnetic field as previously discussed and depends on the input frequency f and pole pitch τ. The thrust F_{x} is then directly proportional to the output power P_{out} and inversely proportional to the speed v_{s} so that

$$F_{\mathrm{x}} = \frac{P_{\mathrm{out}}}{v_{\mathrm{s}}}, \tag{2}$$

which is the case for every linear-motion electric motor. (Gieras, Piech and Tomczuk, 2011)

1.2 Linear positioning systems

Linear motor generally refers to the electromagnetic components of the system, in other words the actual motor, and not the other parts of the system. However, a linear motor is usually part of a larger system, for example an automation system. For a fully functional linear positioning system, many other components are also needed. These include a mounting frame, magnet track, bearings, linear guides to support the rotor, positioning system that consists of the motor, controller, linear encoder and the feedback to the controller, safety end bumpers and limit switches to stop the movement and prevent accidents and the cable management system to provide and manage the cabling to the moving unit. When these other components are included, the system can be generally referred as a positioning stage or system. All of the components together make up a system where we can simply supply it power and either position, torque or velocity information and it will perform as required.

1.2.1 Control of linear motors

Similar electromechanical principles apply to rotary and linear motors. In most cases we can use the same control techniques, controllers and converters with a linear motor as with a similar rotary motor. But to get the most out of a linear motor and make use of all the possibilities a specifically developed controller for a linear motor is required.

The components needed to control and position the linear motor stage according to reference values are the controller and amplifier, measuring and feedback system, power supplies and cabling with a cable management system. Additionally, limit switches may be used at the end of the tracks to stop the motor in case of malfunction or this functionality can also be implemented electronically with the measuring system. (Aerotech Inc, 2010)

A linear encoder is usually optical or magnetic. Optical encoders are more common and can reach higher levels of accuracy than magnetic encoders. The maximum accuracy of a magnetic encoder is around $\pm$ 3 μm/m while a linear encoder can reach accuracies up to $\pm$ 1 μm/m. Many factors can reduce the obtainable accuracy, for example vibrations, flatness of surfaces, position of the encoder and thermal factors. (Hagl and Heidenhain, 2000)

The terms drive and frequency converter are often confused and used interchangeably. A (linear) motor drive refers to the complete positioning system including all the necessary components. A frequency converter or a converter is simply the electrical controller which contains the control logic and power electronics. The inputs to the converter are the power supply, feedback from the encoder system and the reference value in either analog form or through a fieldbus for example. The converter then controls the motor to move to the desired position or give the desired amount of torque.

1.2.2 Resolution, accuracy and repeatability

Resolution, accuracy and repeatability are terms describing the precision of the movement of the linear motor stage. They are often confused and can be defined in many ways.

Resolution is the smallest mechanical movement step that the linear positioning system can make. It depends on the resolution of the encoder, the type of the drive system and the bearings. In an encoder the resolution is the smallest movement step that the encoder can measure. (Xu, 2016)

Accuracy or uncertainty in linear motor systems is affected by the feedback mechanism, the type of the drive system and the bearings. When moving to a desired position in the movement area, accuracy is the largest difference between the actual position in space and the position measured by the encoder system.

Repeatability may be thought of as the range of positions when a system is repeatedly positioned to the same location in uniform conditions. Uni-directional repeatability ignores the effect of backlash and hysteresis in the system and is measured by approaching the point from one direction while bi-directional repeatability takes account both directions. In other words, repeatability is the ability of the system to be servoed to the same position consecutively where the largest error between the successive attempts is the repeatability. (Aerotech Inc, n.d.)

Figure 1 illustrates the difference between accuracy and repeatability. The first picture from the left shows a situation where both the accuracy and repeatability are low. In the second picture on the right the repeatability is high while accuracy remains low. It can be seen that the consecutive positionings of the stage are close to each other but still far from the desired position. The lowest picture shows a situation where both repeatability and accuracy are high and the desired position is hit every time.

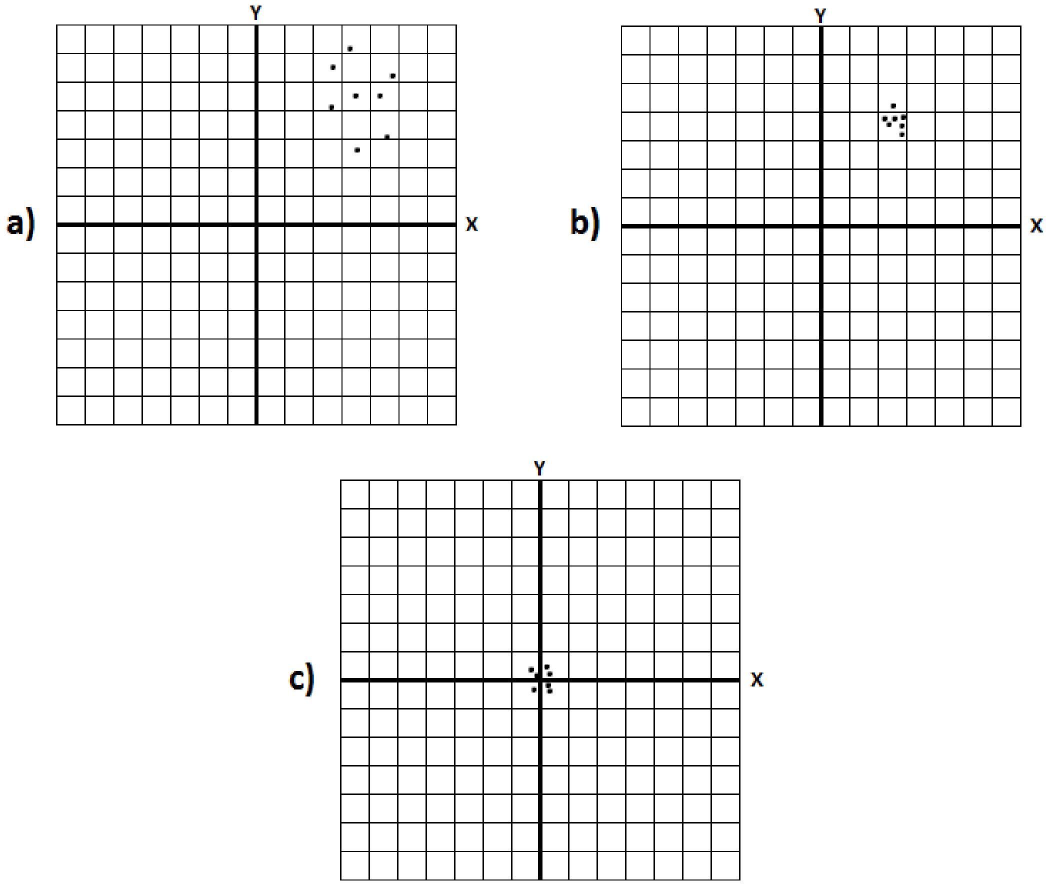


Figure 1. Difference between accuracy and repeatability. The dots indicate the consecutive positionings while the desired position is located in the origin of the XY-coordinate area. Figure a) shows a situation where both the accuracy and repeatability are low. On figure b) the repeatability is high but accuracy remains low. Finally, on figure c) both the accuracy and repeatability are high.

1.2.3 Fieldbus

Fieldbus is a family of standardized industrial computer networks that are used in real-time distributed control. It is a method of connecting instruments and actuators for example at a manufacturing plant. The fieldbus network structure allows many different network topologies. While previously the devices were connected by a current loop or a RS-232 connection, each device needed to have separate wires running between the controller and the device. The advantage of a fieldbus is that the number of cables is greatly reduced as hundreds of signals from different devices can be transmitted digitally in the same wires. (Moore Industries, 2006)

There is a large variety of standardized competing fieldbus technologies. The most common today include CAN, EtherCAT, Foundation Fieldbus, Modbus and Profibus. In our application it is necessary to transfer real-time data between the National Instruments Real-Time controller and the linear motor servo drives. An Ethernet-based fieldbus is a good choice since it is digital, real-time and most importantly compatible with the rest of the system. National Instruments uses the EtherCAT standard in its compatible servo drives.

The EtherCAT (Ethernet for Control Automation Technology) is an Ethernet-based fieldbus system. It is originally invented by Beckhoff Automation in 2003 and it is standardized in IEC 61158. Its synchronization and full bandwidth utilization capabilities make it good option for motion control applications where synchronizing a large number of drives is required. It is possible to achieve very short cycle times and implementation costs are also quite low since no special hardware is required for the master node, although a special chip is required for slave nodes. Up to 65 535 devices can be connected to an EtherCAT network.

The basic function principle of EtherCAT is that the Ethernet packet is not processed at every node. Instead the data is only processed when needed and the frame is not completely received before the processing can begin. The frame is only delayed by the hardware propagation delay times. The master node is the only device that can actively send frames while all the other nodes forward the frames. It uses the MAC (Ethernet Media Access Controller) standard without any additional communication processor. This makes it possible to be implemented on any hardware with an available Ethernet port with a simple software upgrade. Protocol stacks are not used since the protocol is optimized for short cyclic data.

The EtherCAT protocol embeds its frame in a standard Ethernet frame, as can be seen from figure 2. The EtherCAT frame starts with a standard header, which includes the length identifier and the type integer. After that the frame contains the PDOs (Process Data Object). They contain the data for every different node. Finally, the frame contains the working counter, which is dependent on the content of the EtherCAT frame. A node can ensure it receives the frame entirely by using the working counter. (EtherCAT Technology Group, n.d.)

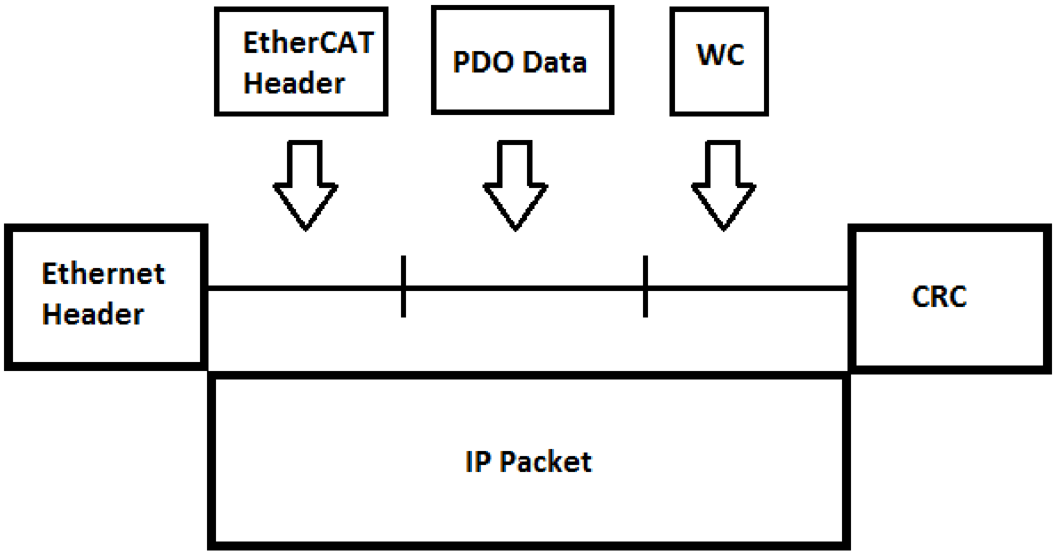


Figure 2. The EtherCAT frame replaces the data portion of a regular Ethernet packet.

The synchronization feature is important in processes where exact simultaneous actions are required, for example a multi-axial servo executing coordinated movements. EtherCAT includes a distributed clock mechanism which allows a low jitter without additional hardware. Each node includes a timestamp in the frame twice, once when receiving the sent message and once when the frame is returning back through the nodes. With this information the master node can calculate the delay for each node.

1.2.4 Multi-axial systems

Quite often movement in one direction is not sufficient. Many applications may require multi-axial movement, such as assembly robots, laser processing systems, material handling systems, pick-and-place machines and so on. There are many different types of configurations and systems to accomplish the desired movements. The movement can be implemented by moving the workpiece, the tool or both. A stage which moves in a two-dimensional horizontal area is called a XY-stage. Z-axis movement refers to vertical movement, so if the stage moves in a three-dimensional space, it is called a XYZ-stage.

The simplest way to accomplish XY-movement is to assemble two single-axis positioning systems on top of each other. There are some problems with this setup that need to be overcome. The lower positioning stage has to be able to move the entire upper positioning system in addition to the load, so it may need to be more powerful than the upper system. In other words, the two axes may not be identical since the upper system only has to move the load. In addition, the accuracy and repeatability may be reduced since the track of the lower positioning system may not be completely straight and may cause deviations in the other axis. Figure 3 shows a common linear motor XY-stage system. Bellows can also be used as shown in the figure to protect the tracks and the encoders from dust and dirt.

Figure 3. A typical XY-stage from Akribis. (Xu, 2016)

If the processing tool is moved instead of the workpiece in a linear movement, the system is called a Cartesian coordinate robot. It can be a two- or three-axis system. The different axes are attached to each other. Figure 4 shows an example of a Cartesian robot.

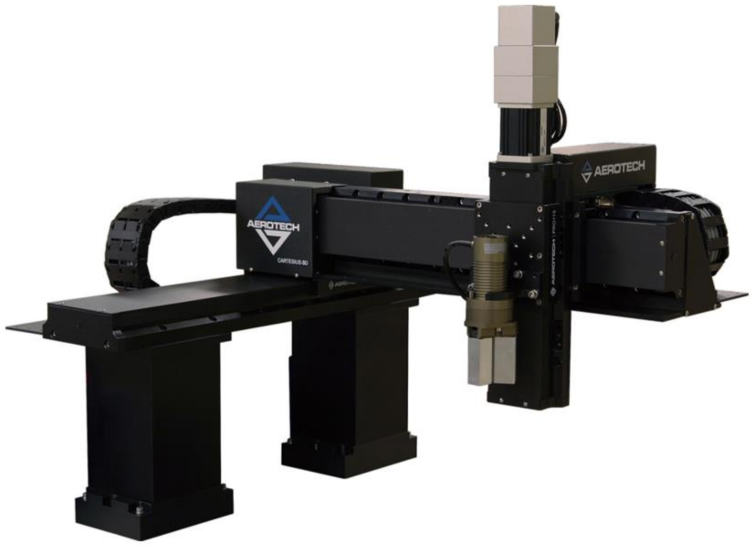


Figure 4. A Cartesian coordinate XYZ-robot. (Aerotech Inc, 2010)

The robot is only supported from one axis and the other axes are connected to it and supported from one end as seen in the figure. Cartesian robots have a working envelope of a two- or three-dimensional rectangular box depending on the number of axes. Typical applications include pick-and-place robots, assembly and dispensing systems. (Brumson, 2001)

Cartesian coordinate robot which is supported at both ends is called a gantry system. It is a special type of Cartesian robot and the benefit is that the dual supports minimize deflection along each axis. They are best suited for applications where large masses need to be positioned with high accuracy. Similar to Cartesian robots, the working envelope is a two- or three-dimensional box. However, the box is usually enclosed inside the supports. As the robot is supported from multiple positions it can lift very large masses with the right construction. (All on Robots, n.d.)

1.2.5 Vertical movement

Because linear motors are frictionless, they are not well suited for vertical applications since a force is constantly needed to overcome gravity even if the load is not moving. In addition, if the power is lost suddenly, the load will fall down pulled by gravity. However, different methods can be applied to overcome these problems. A counterweight can be used giving the advantage that the force needed to move the load against the gravity is reduced and in case of a power loss, the mass is returned to the center instead of falling down. Figure 5 shows an example of a vertical linear motor application with a counterweight. Another option is to use a spring system where the spring compensates the gravity force.

Figure 5. Example of a Z-axis linear motor positioning stage with a counterweight mechanism. (Aerotech Inc, n.d.)

Unless very high accuracy is needed it is in many cases advisable to use a ball-screw or a lead-screw system for vertical movement as their design removes the back-driving in case of power loss and the need for complex counterweight mechanisms.

1.3 LabVIEW

LabVIEW is a development environment developed by National Instruments. It uses a visual programming language and is very popular in industrial automation, testing automation, instrument control, data acquisition and monitoring systems. LabVIEW is used to implement the program developed in this thesis.

The LabVIEW graphical language is named “G” and it is a dataflow programming language. The programmer connects functional nodes using wires which transfer variables and data. Any function or node will execute as soon as all input data is available. In other words, G is capable of parallel execution. The wire colors indicate the data type being transferred. Figure 6 shows a simple example of the programming language where the user can enter coordinates and supply them to a function block. Green wire color refers to a boolean value while orange color refers to a numeric value. Orange color is a floating point value while blue would be an integer value.

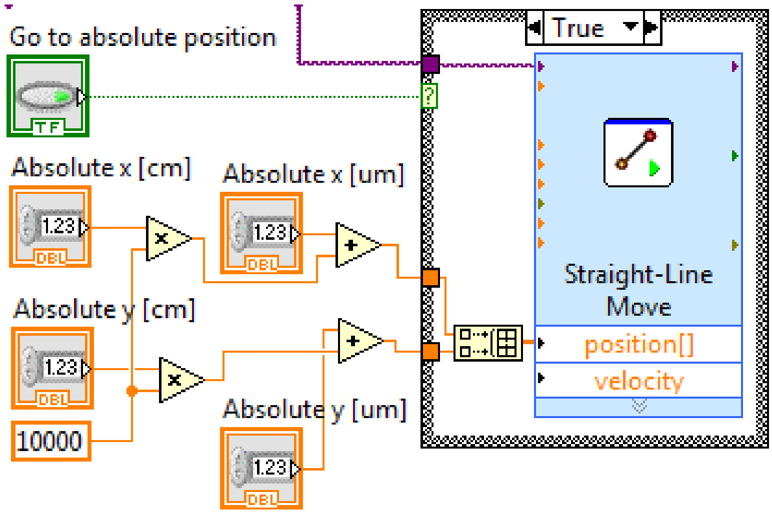


Figure 6. An example of the G programming language.

Each LabVIEW program consists of the user interface (the front panel), the block diagram (the back panel) and the connector panel. The front panel is the panel normally visible and acts as the user interface where all the controls and inputs are available for the user to supply information and the indicators that acts as outputs supplying information to the user. The back panel or block diagram is where the actual programming is done with G and it contains all the structures and functions to perform the desired operations.

1.3.1 Modules

The capabilities of LabVIEW can be expanded with modules that are acquired separately. They may add additional features and enable connectivity to different programs and interfaces. To develop the program for the laser processing system, the Real-Time Module is necessary as it allows developing deterministic real-time applications for the motion control system. It also enables communication with the embedded real-time computer which is used to communicate with other devices in the laser processing system. In addition, the SoftMotion module is used as it allows developing custom motion control applications with graphical user interfaces and gain access to trajectory generation and spline interpolation and other high-level interfaces.

1.3.2 SoftMotion module

The SoftMotion module is the most important module to be used when developing the movement for the linear motors in this thesis work. It makes possible to configure motion axis settings in LabVIEW, tune the servo motors as necessary, program motion profiles and create automated motion applications.

Figure 7 shows a simplified diagram of a common positioning system. The SoftMotion module acts as the motion controller part between the user application and the selected communication method with the amplifier and motor. The user application is LabVIEW and the communication method EtherCAT fieldbus in this case. The motion controller receives the high-level commands from the user application, converts them and calculates command signals used by the amplifier to move actuators. These commands are then communicated via the fieldbus. In addition, the motion controller monitors the system for faults and asynchronous events and acts according to the received events.

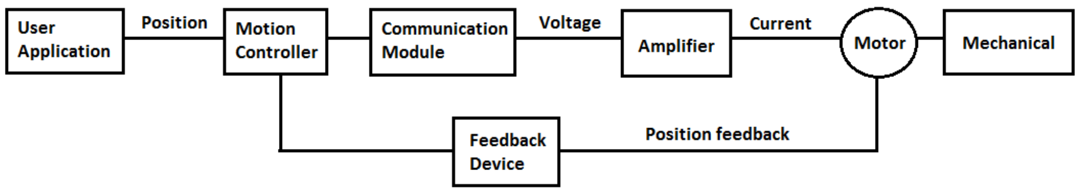


Figure 7. A general motion system diagram for position control.

The different motion controller components and their general operation is shown in figure 8. The supervisory control performs command sequencing and coordination including fault handling. It also handles system initialization such as homing to a zero position. It receives the commands from the user to start or stop movement and passes them on to the trajectory generator while taking account the events coming from the I/O-pins. The trajectory generator then creates set points for the control loop based on the received move constraints. These constraints include maximum velocity, maximum acceleration and deceleration and maximum jerk that the mechanics can tolerate. The control loop runs at a much higher rate and includes position and velocity loops. It receives the data from trajectory generator, reads the feedback from encoders and generates the command signal to be passed on to the amplifier.

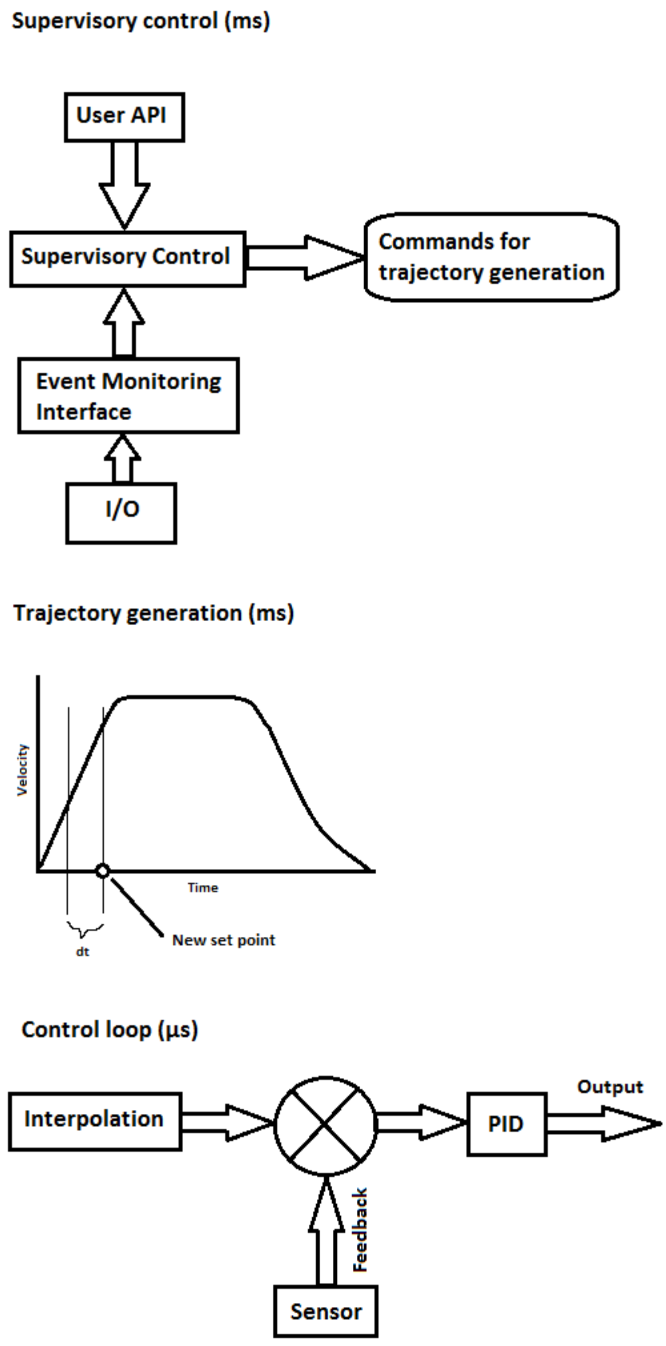


Figure 8. The fundamental components and the operation of the SoftMotion module.

When using SoftMotion, the drives used first have to be registered as axes. In addition, when a multiaxial system is used with two- or three-dimensional working area, a coordinate space can be created that combines the axes and allows performing coordinated moves in a multidimensional area. SoftMotion has function blocks to perform certain moves that are typical to any motion control system. These include straight-line moves, arc moves and contouring.

Straight-line move performs a move in a straight line between two points in space. The coordinates can be expressed as absolute or relative and the velocity, acceleration and jerk can all be defined separately. Arc move causes a move in a circular, spherical or helical path. The angle, radius, velocity acceleration and jerk of the move can be defined. Contouring is a move where the user specifies a set of coordinates and the spline engine generates a trajectory using cubic spline algorithm.

The functions can be used in both synchronous and asynchronous modes. In synchronous mode the program or loop waits for the function to finish before continuing. In other words, the executing of the LabVIEW loop hangs until the requested move is complete. On the other hand, in asynchronous mode the execution of the move is started on the rising edge of a boolean input value and does not interrupt the rest of the program. The user has to implement a way to manage the order of execution of the program and also handle timeouts and warnings. The functions provide boolean outputs which can be used to determine when the move is running or finished.

1.4 Laser processing tasks in solar cell manufacturing

Solar panels are panels engineered to receive sun’s rays energy and convert it into electricity. A photovoltaic (PV) or solar cell is what solar panels are constructed from. They convert the light’s energy to electric power by employing the photovoltaic effect. The operation of a solar cell requires light (sunlight or artificial), the separation of the charge carriers of opposite types and the connections to external circuit to transfer the energy to.

Photovoltaic effect converts light (photons) to electricity. Quite many materials can in theory be used for photovoltaic conversion, but in practice semiconductor materials are used in the form of a p-n junction. First generation solar cells consist of an antireflection coating on top with the front contacts. Under the top layer is the actual semiconductor material, emitter-base junction and at the bottom side is the rear contact. (Honsberg and Bowden, n.d.)

The generation of the current involves two processes. First, the photons (in other words light) are absorbed by the cell creating electron-hole pairs. They will be generated if the photon has an energy greater than that of the band gap. After that, a second process prevents the recombination of the electron-hole pairs with the p-n junction by separating them spatially. This is accomplished by an electric field in the p-n junction. Thus if the emitter and base of the solar cell are then electrically connected through a load, the electrons will flow through it and generate electricity. (Honsberg and Bowden, n.d.)

Solar cells may be divided into first, second and third generation cells. The first generation cells are the traditional solar cells and are typically made from crystalline silicon. They are quite thick and bulky compared to the second generation cells but often have the best efficiency even today with all the other cells available. The second generation cells are also called thin-film solar cells because they use layers of semiconductor materials which are only a few micrometers thick. This allows them to be flexible and enables many new installation possibilities. The third generation cells include a number of emerging and upcoming technologies that most of have no commercial benefit yet. They include solar inks, solar dyes and conductive plastics. (NREL, n.d.) With the developed laser processing system, the focus will be on the second generation solar cells and their scribing.

1.4.1 Thin-film solar cells

The second generation solar cells are based on Copper-Indium-Gallium-Selenide (CIGS), amorphous silicon or Cadmium Telluride (CdTe). As they only use a thin layer of semiconductor materials, the cells can be built flexible. In addition, the combination of using less materials and lower cost of the manufacturing process should allow producing and selling the panels at a much lower cost than first generation cells. (Solar Tech USA, 2014)

The structure and layering of the thin-film cell is always the same even if different semiconductor materials are used. A thin semiconductor film to absorb solar energy is positioned between two conducting contact films, which allow the electric current to flow. At the bottom or top depending on the configuration is the substrate to which the films are deposited. The front-side conductor has to be transparent while still conducting electricity, and transparent conducting oxide (TCO) materials for example indium doped tin oxide are used. (Bovatsek et al., 2010)

Figure 9 illustrates the structure of a typical CIGS cell. The substrate is typically soda-lime glass as it has been shown to increase open-circuit voltage. The back contact is a Molybdenum (Mo) metal layer which conducts electricity while reflecting unabsorbed light back into the CIGS material. On top of that is the p-type semiconductor material, in this case CIGS. On top of this absorber, a thin n-type buffer layer is added. This is usually cadmium sulfite (CdS). Finally, on top the zinc oxide (ZnO) layer acts as a transparent conducting oxide layer to conduct electricity. (Energy.gov, n.d.)

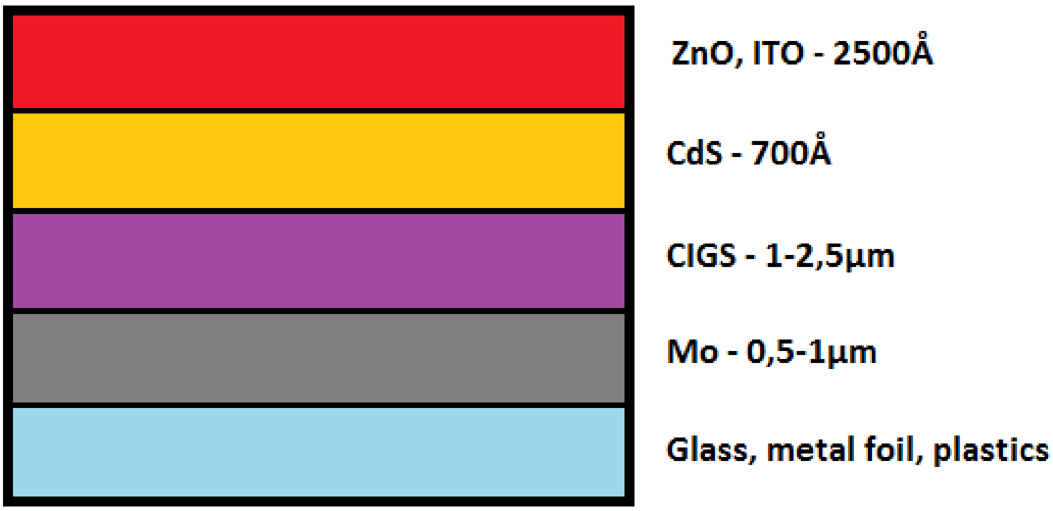


Figure 9. Structure of a CIGS cell.

The solar cell can be assembled in either substrate or superstrate configuration. Figure 10 shows the superstrate configuration of a thin-film photovoltaic cell. Here the substrate faces the sunlight and thus has to be transparent to pass the light through to the absorber.

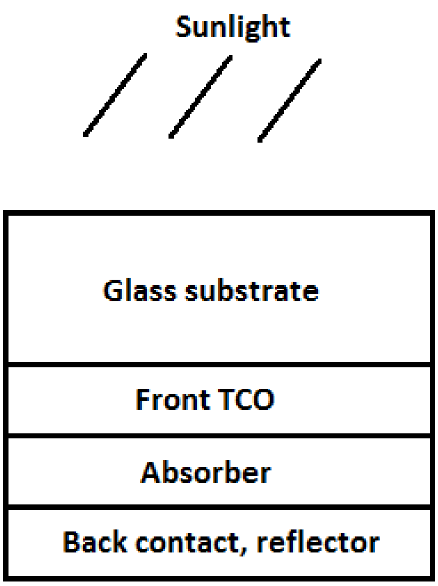


Figure 10. Superstrate configuration.

In a substrate configuration as shown in figure 11, the other films are deposited on top of the substrate. The substrate can then be opaque material since it is under all the other films. (Litmanen, 2015)

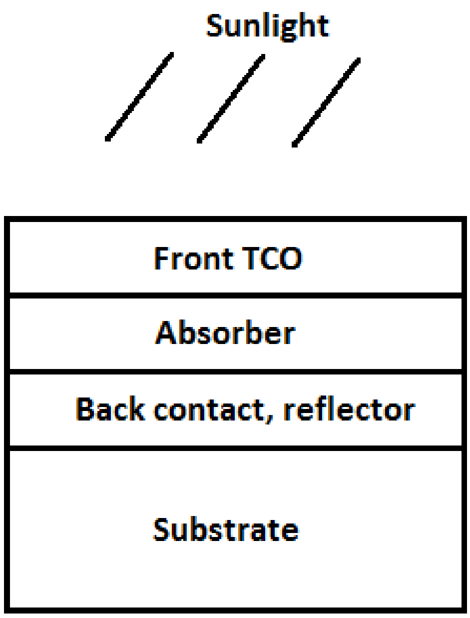


Figure 11. Substrate configuration.

1.4.2 Scribing of solar cells

In thin-film photovoltaic devices, the most important laser process when manufacturing the cells is thin film removal, in other words scribing. It is important so that we can have electrical separation between the different parts of the serially connected cells. Laser scribing is a cost-effective and accurate method compared to mechanical scribing while it also has better scribe quality and faster processing times. The process includes irradiation of the glass panel by employing the laser beam. This results in the removal of some of the thin film layers. When manufacturing second generation cells, three scribes are necessary and are referred to as P1, P2 and P3, as seen on figure 12.

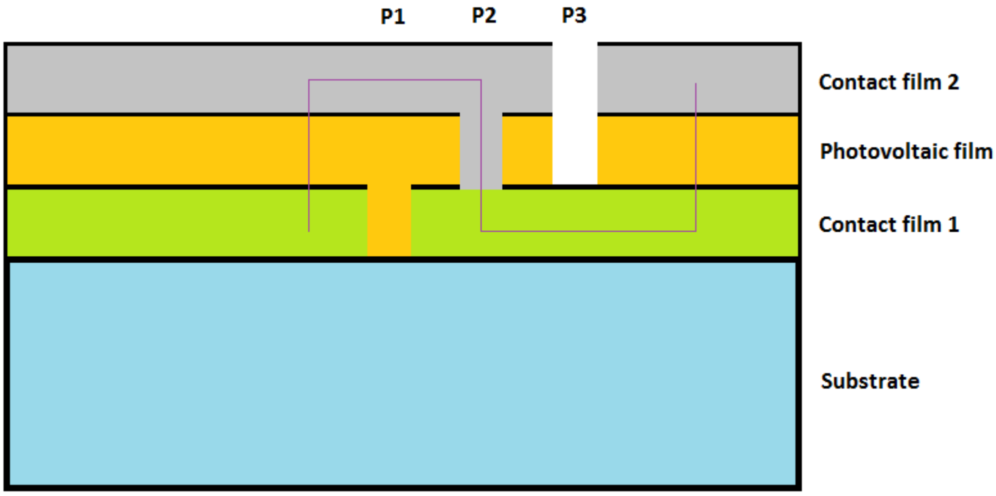


Figure 12. The structure of a TFSC and the different scribes required to make the current flow as shown with the purple line.

These three scribes make up the structure of the solar cell where electricity has only one path to flow, through the photovoltaic film and back to the other contact film. The current flows back and forth multiple times and it results in series connection of the cells and allows for more efficient device operating at higher voltages and at lower current. The P1 scribe is performed to erase the electrical contact film from the glass substrate. This is required to be able to separate the cells electrically. The P2 scribe is then used to remove part of the PV film so that the contact films can connect through it and allow electrical current to return back to the other contact film after it has passed through the photovoltaic film. The P3 scribe finally removes both the PV film and the second electrical contact film and separates the cells in the upper contact film electrically.

Usually, it is preferred to do the scribing from the substrate side of the cell. The efficiency is higher and thermal effect lower compared to film-side scribing. When doing the scribing from the substrate-side, it is not necessary to melt and evaporate all of the film but the removal can be induced by the stresses and heating of the absorption point.

In film-side scribing on the other hand, material removal usually happens by melting and evaporation. Therefore, accurately creating sharp edges of the scribing grooves is challenging. In addition, an insulating layer might be necessary to add on substrates conducting electricity. The parameters of the laser such as the pulse duration, repetition rate, power of the laser and the marking speed must be very accurately set and optimized to achieve this and not damage the other layers in the process. (Litmanen, 2015)

As CIGS modules are used in substrate configurations, it means that the P2 and P3 scribes cannot be easily done from the substrate side as the first layer is opaque. The P1 scribe is usually done using nanosecond laser pulses from the film side. It is also very challenging since the laser parameters need to be optimized correctly as discussed earlier.

1.5 Color marking

It is possible to generate different colors on metal surfaces with laser pulses. This process is called laser color marking. It is based on an oxidation process and thin film effect following it. The thickness of the oxide layer generated by the laser pulses defines how light is reflected off the surface and therefore generates different colors. Depending on the oxide layer quality and thickness, the color may change when examined from different viewing angles. The quality and uniformity of the color and the shade depends on the laser parameters such as pulse duration, repetition rate, laser power, marking speed, line spacing and the focal spot diameter. (Laakso, Ruotsalainen, Pantsar and Penttilä, 2009)

One future research topic with the laser processing system is color marking. The vertical movement of the scan head is developed with this future research in mind as it makes it possible to adjust the focal spot diameter. The other necessary parameters of the laser can be changed using the developed laser control interface or other settings in the program. Some preliminary tests were also conducted at the end of this thesis.

2 LASER PROCESSING

Typical laser processing system consists of the laser source and optical systems to process and modify the laser beam as desired. In addition, if the laser is required to do precise tasks like scribing, marking or cutting, a beam deflection and positioning scan head can be used to position and focus the laser beam. Moreover, some kind of a motion control system is used to move either the laser beam or the workpiece or both. All of these devices are then controlled together with a control system which also includes monitoring of the laser beam via a spectrometer or a camera system for example.

Common laser processing applications can be divided in removal of material, sealing of components like welding for example and surface improving methods. These are illustrated in figure 13. (Sigmakoki, n.d.)

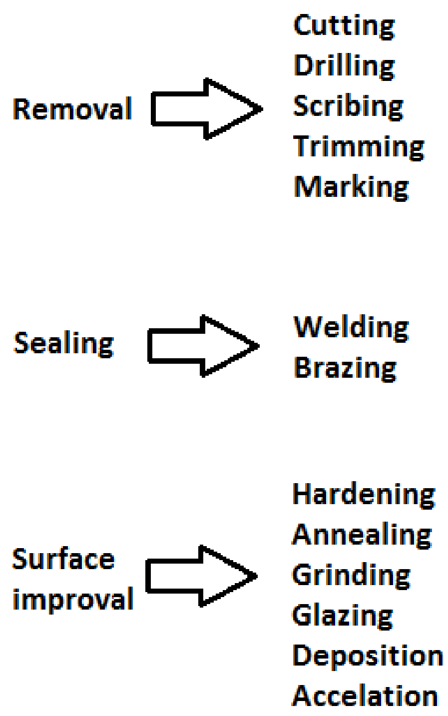


Figure 13. Different laser processing applications.

2.1 Laser processing system

The laser processing system in the laboratory consists of a laser source, scan head, linear motor XY-platform and a monitoring system which consists of a camera and spectrometer. In the future also an optical system will be added to control and modify the laser beam entering the scan head. The system is controlled using a desktop PC and a National Instruments PXI embedded real-time controller as illustrated in figure 14. The PC has an RTC4 interface card that is used to interface with the scan head and laser. The laser can also optionally be controlled via a regular serial port. The embedded controller is connected to the PC using a regular Ethernet connection. It is then connected to the camera and spectrometer via special interface cards. The linear motor system is driven by servo controllers which communicate with the embedded controller via EtherCAT.

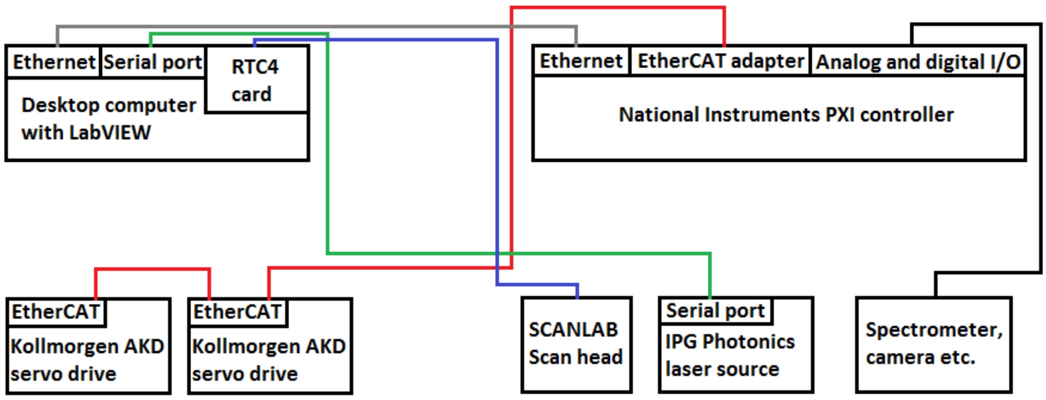


Figure 14. The illustration of the data communications in the system after the completion of this thesis work.

Figure 14 shows the different components of the system and connections between them. Gray color refers to a regular Ethernet connection, red color to an EtherCAT fieldbus and green to a serial port connection. Blue and black colors are the custom connections with the scan head and the I/O pins.

The focus of this work is to develop a control system using LabVIEW and other needed software that combines the laser, scan head and linear motor systems and controls them together in sync so that they can perform scribing- and other desired tasks. In other words, LabVIEW programs first need to be developed for all the different components to be able to interface them with LabVIEW and then combine all of them into one complete system and program the required functionality using LabVIEW.

In addition, the first part of the work was to research the available options on the market and order an adequate motion system to move the workpiece, since the old system in the laboratory was not functioning properly.

2.1.1 Laser

The laser is an ytterbium fiber laser from IPG Photonics. It is a pulsed mode of operation laser with selectable pulse duration and random polarization. The common optical characteristics of this laser are listed in table 2.

Table 2. Optical parameters of the laser. (IPG Photonics, 2009)

Characteristic	**Symbol**	**Min**	**Typical**	**Max**	**Unit**
Pulse duration	T1-T8	4, 8, 14, 20, 30, 50, 100, 200			ns
Central emission wavelength	λ	1055	1064	1075	nm
Emission bandwidth	$\Delta\lambda$		5	10	nm
Nominal average output power	P_{nom}	19	20	21	W
Output power adjustment range		10		100	%
Extended pulse repetition rate	RR	1,6		1000	kHz
Maximum pulse energy			1		mJ
Maximum peak power			15		kW
Laser switching on/off time			2	3	µs
Long-term average power instability			2	5	%

The laser beam is transferred via an optical fiber cable to the output head, which is then connected to the scan head. The laser can be controlled via a special digital control interface that uses a DB-25 plug connector and connects to the RTC4 card. Another option is to use the common RS-232C interface that can be connected to a regular serial port. The parameters of the laser can be read and written and the laser and guide laser enabled and disabled via the remote interface. (IPG Photonics, 2009) Previously the laser has been controlled with an external application but now the goal is to make it possible to control it in LabVIEW environment.

2.1.2 Scan head

The scan head is a hurrySCAN II 14 digital scan head from SCANLAB with a 100 mm objective. It deflects the laser beam using two rotating mirrors and galvanometers so that the laser beam can be positioned anywhere in a two-dimensional working plane. It also has a lens which orients the laser beam in such a way that it always hits the scribed area from top-down and not in an angle. This allows making markings and scribes with the laser in a small working area.

Physically the scan head is shown in figure 15. The laser beam is entering the head from the side and is projected down through the scan lens. The scan lens is focusing the beam on to the working plane and thus the plane must be at a specific distance away from the lens.

Figure 15. The scan head and its lens. (SCANLAB, 2011)

Additionally, a dynamic focusing system positioned in front of the lens could be used to change the focus electrically. To achieve optimum reflectivity at the mirrors, they have a coating which is dependent on the wavelength and power of the laser. This means that a different scan head must be used for different wavelength lasers. The scan head has power and communication connectors. It is connected to the desktop computer via the RTC4 interface card. (SCANLAB, 2011)

2.1.3 RTC4 card

The card has a special DLL file which allows external programs to communicate with the scan head. This DLL is going to be used to control the scan head with LabVIEW. Figure 16 shows a general diagram of how the scan head and other parts of a common laser processing system come together.

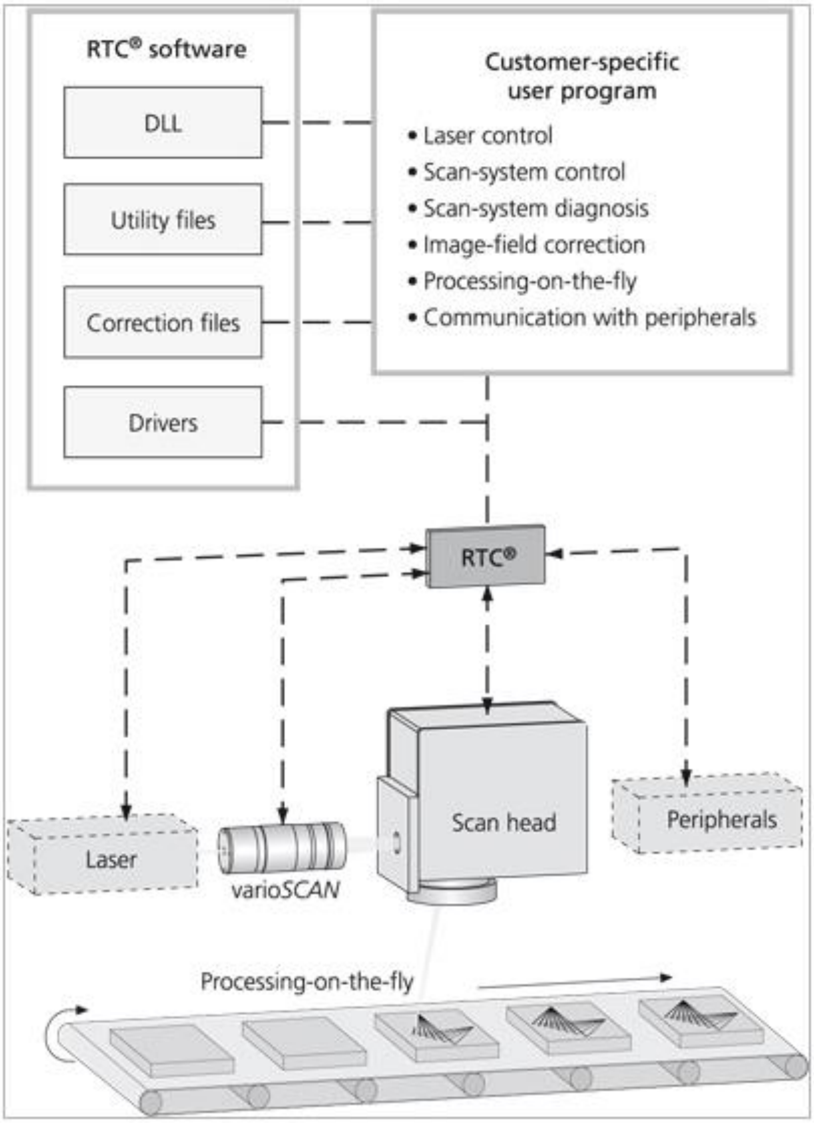


Figure 16. The structure of the communication between the RTC4 card, software and the equipment. (SCANLAB, 2009)

The scan head could also be controlled separately without the RTC4 card. However, the two-mirror system in the scan head causes distortion of the image field. This happens due to the distance between the mirror and the image being dependent on the scan angles as seen on figure 17. In addition, a distance unit in the image field of the scan head is not directly correlated with the scan angle of the mirrors. Instead, it is proportional to the tangent of the scan angle.

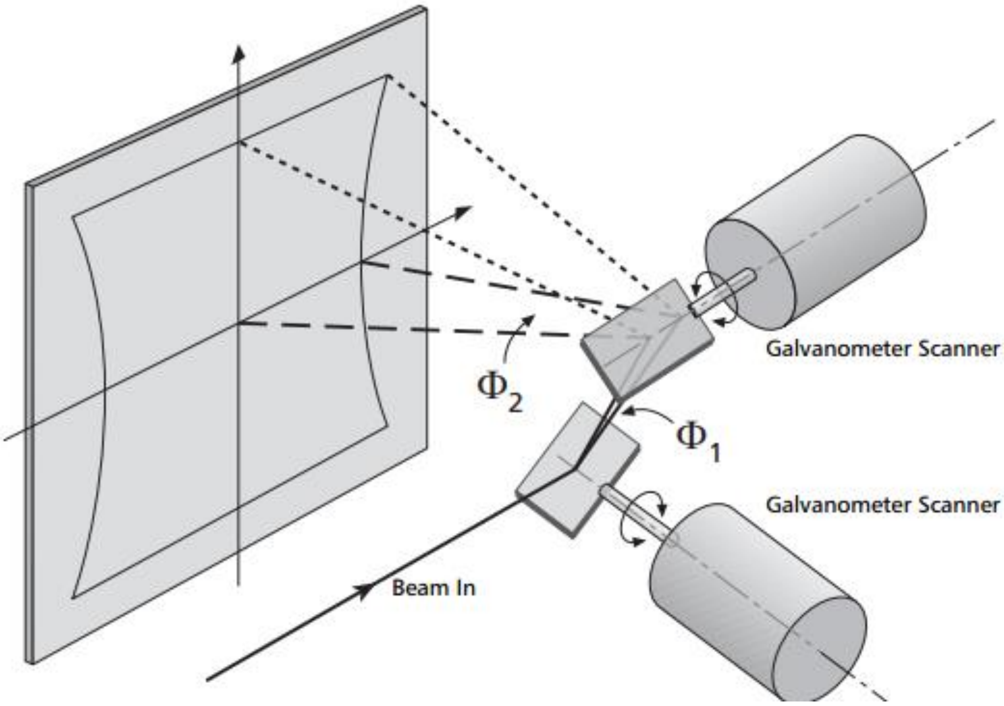


Figure 17. Distortion of the image field in the scan head. (SCANLAB, 2011)

This needs to be compensated in some way to achieve desired performance. The RTC4 board provides a correction algorithm to compensate for this field distortion. It is based on a correction table and is created individually for every system and it compensates the distortions automatically so one can simply give the board absolute or relative coordinate instructions without having to worry about the distortions.

2.1.4 Motion system

The previous linear motor system in the laboratory was not functioning properly and it was custom manufactured so there weren't any instructions or manuals available for it. In addition, it had a simple analog control with no fieldbus communication possibilities which would have made developing the controller to work with the rest of the system very difficult. Therefore, a new motion system had to be researched and ordered first.

A linear motor system was the only reasonable choice for this application, since other linear actuation methods such as ball screw or belt and pulley are simply not accurate enough, as discussed in chapter 2. Micrometer level accuracy and repeatability is required from the system, since the scribes in the solar panels must sometimes be positioned very accurately and must be at an exact distance between each other (which requires high repeatability). In addition, linear motors depending on the model can have very high acceleration and velocity, while still maintaining high repeatability.

This enables faster processing and movement of the workpiece if necessary. Moreover, since the laser processing system must be controlled as a complete system and we want to use high-level commands to communicate with the servo drives, fieldbus connectivity is a necessary feature.

One possibility is to select all the drive components separately, including the linear motors, encoders, servo controller, mounting frame and cable management systems. However, since the focus of this work is more in the development of the control system as a whole and the motion control of the system, it was decided to select a complete assembled system.

After a thorough research of the market, Akribis linear motors with AKD Kollmorgen servo drives were found to be the best option from all of the available choices. Akribis has a good linear motor multiaxial stage as seen in figure 8 which can be equipped with an encoder that enables high accuracy and repeatability. It is scientific, accurate, affordable and best suited for the application. They use an ironless linear motor which has no attractive forces or cogging and is best suited for smooth velocity control which is just what we need. The electrical parameters of the motors are shown in table 3. As can be seen from the table, the motors are not identical. Instead, the lower motor is more powerful since in addition of the actual load it has to move the entire upper linear motor system as they are positioned on top of each other.

Table 3. Electrical parameters of the linear motors.

Performance parameters	**Axis**	Top	Bottom
	Motor	AUM4-S2	AUM4-S3
Continuous force, coil 100 °C	N	110	166
Peak force	N	624	936
Motor constant	$N/\sqrt{W}$	15,8	19,4
Continuous power	W	48,7	73,0
Peak power	W	1555	2332
Electrical cycle	mm	60,0	60,0
Max bus voltage	V	330,0	330,0
Max coil temperature	°C	125,0	125,0
Thermal dissipation constant	W/°C	0,65	0,97
Continuous current	A_{RMS}	2,3	2,3
Peak current	A_{RMS}	13,0	13,0
Force constant	N/A_{RMS}	48,0	72,0
Back EMF constant	$V_{peak}/(m/s)$	39,2	58,8
Inductance	mH	7,00	10,50
Terminal resistance, 25 °C	Ω	9,20	13,80
Electrical time constant	ms	0,76	0,76
Mechanical parameters			
Coil mass	kg	0,56	0,89
Coil length	mm	121,0	181,0

The repeatability of the system is about a few micrometers as can be seen in table 4 depending on the encoders and the how their signals are processed. The linear encoders are Renishaw optical readheads which use a reflective tape to acquire the position.

Table 4. Performance parameters of the selected linear motors.

Specification parameter	**Unit**	**Value**
Straightness	µm	±3 µm / 25 mm NTE ±10 µm / 300 mm
Flatness		±3 µm / 25 mm NTE ±10 µm / 300 mm
Repeatability (1 µm resolution)		±3 µm (40 µm scale pitch)
Repeatability (0,5 µm resolution)		±1,5 µm (20 µm scale pitch)
Repeatability (0,1 µm resolution)		±1 µm (20 µm scale pitch)
Repeatability (analogue)		±5 counts
X-Y orthogonality	Arc-sec	10

In addition, the straightness and flatness of the tracks are important to not be too large, since the focus of the laser beam depends on the distance between the scan head and the workpiece. And so if the distance between them varies too much while the workpiece is moved, the laser is not functioning at the same effectiveness all the time.

The AKD series servo drives are from Kollmorgen. Kollmorgen has been in partnership with National Instruments since 2013. The combining of National Instrument motion control software and Kollmorgen drive and motor technology enables powerful tools to build complex machinery and simplifies the design greatly. (National Instruments, 2013) The model selected is shown in figure 18. It has EtherCAT and CANopen connectivity and includes a position indexer. In addition, the controller has digital I/O pins of which some are used for the limit and home switches and some can be used for other future projects.

Figure 18. AKD servo controller with EtherCAT and CANopen connectivity. (Kollmorgen, n.d.)

LabVIEW is used to control the entire processing system as a complete setup and connect all the devices together. The processing system has a Real-Time PXI controller PXIe-1071, which allows communicating with the EtherCAT fieldbus to the linear motor drives. The Real-Time controller is then connected to the desktop computer via a regular Ethernet connection. The monitoring of the laser is also performed via the controller with a spectrometer and a camera.

For the vertical movement, there was a ball screw linear actuator from Neff-Wiesel available. Its maximum speed is 1 m/s, repeatability $\pm 0{,}05 \text{ mm}$ and maximum load 650 N. The technical specifications are sufficient for moving the scan head up and down for focusing of the laser beam. Thus, a compatible servo drive had to be ordered, and it was also selected from Kollmorgen. The servo controller is the same that is used in controlling the linear motors, so it can easily be communicated via the EtherCAT fieldbus. It is designed to be used with many different types of motors, including linear motors and also the servo motor that is selected for the vertical movement. The selected motor is a synchronous servomotor with a holding brake and has a inductive single turn resolver.

3 DEVELOPMENT OF THE CONTROL SYSTEM

The goal was to develop a program in LabVIEW environment that combines all of the components of the system such as the laser, scan head, movement of the linear motor stage and vertical movement of the scan head, enables them to be controlled from the same environment and allows them to work in synchronization with each other. For example, when a shape is drawn on the drawing area of the program, the scan head and linear motor should move and the laser turn on and off in sequence so that the desired scribe is achieved on the workpiece.

The working area of the scan head, 54 x 54 mm, is much smaller than the linear motor movement area, 29 x 29 cm. One of the goals is also to enable us to process larger workpieces in comparison to using only the scan head. For example, an algorithm should be developed that allows to scribe a shape using the scan head, move the workpiece to a different location, and scribe again. In addition, another option in achieving a larger scribing area is to scribe by keeping the laser stationary and using the linear motor stage to move the workpiece and draw the larger, desired shapes. However, before developing the actual program to perform the scribing, the interfaces and communication modules need to be implemented for the different components of the system in LabVIEW environment, since previously separate programs were used to control different components of the system.

3.1 Laser communication interface

The communication to the laser source can be achieved via a serial port or the RTC4 card with a special connector. To be able to control the laser in LabVIEW, the serial port option must be used. LabVIEW has a VISA (Virtual Instrument Software Architecture) library which allows communication using serial ports. (National Instruments, 2011)

The manual of the laser source lists all the possible commands that can be used to interface with the laser. They include getting information from the laser, such as the current setting values, temperature of the laser, state of the power supplies and so on. Another set of commands enables writing the setting parameters and enabling and disabling the guide laser and the real laser emissions. The goal is to be able to set and read the parameters such as the operating power of the laser, pulse repetition rate and pulse duration and enable and disable the guide laser and laser emissions. The guide laser is a simple laser pointer which allows seeing where the scan head is positioning the beam.

When starting up the program, the communication port to which the laser is connected first needs to be opened for use. This is done as seen on figure 19. The required settings such as the baud rate, data bits and parity are described in the laser source manual.

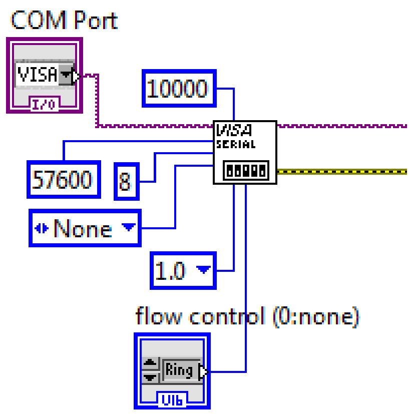


Figure 19. Opening the communication port for use in LabVIEW. The serial port can be selected with the purple selector, which is visible on the user interface. The blue wires carry integer values from the constants such as the baud rate, data bits and parity to the VISA Serial -block. This block then opens the port and transmits a handle for it via the purple line to be used elsewhere in the program.

After the port has been opened for use, commands can be sent using LabVIEW. For example on figure 20 the laser and guide laser toggling on and off commands are shown. The desired command is put inside a case structure which is connected to a button. When the button is pressed or enabled remotely using local variables, the case structure value changes to true and the command is sent. At the end of the wires that go through all the case structures is a block which actually writes the commands to the port and also receives data if needed.

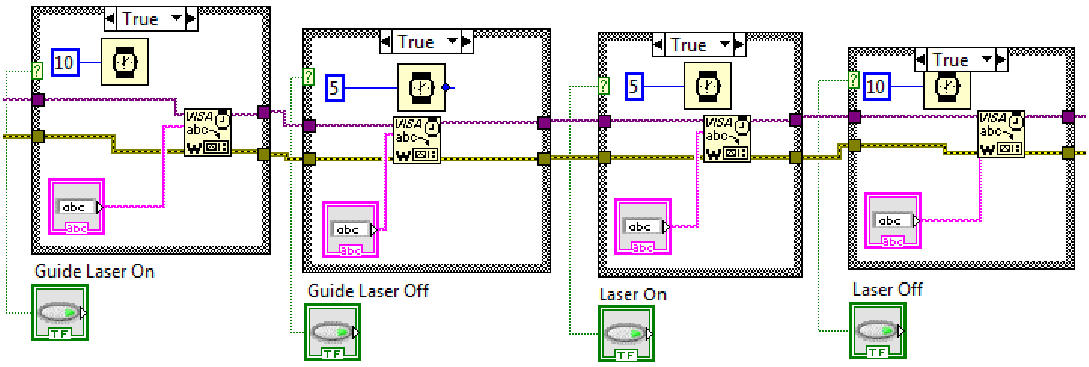


Figure 20. Sending commands using the serial port.

The user interface to control the laser is shown on figure 21. It has fields where the operating power, pulse duration and pulse repetition rate can be specified and then a button which sends the commands to the laser. For the laser and guide laser control there are four buttons. First button completely enables and disables all laser controls and the laser. Second button decides whether the guide laser or the actual laser is going to be used. Normally the laser is controlled automatically from other parts of the program which scribes the shapes (as long as control is enabled). However, if for some reason the laser has to be manually toggled on and off for example for testing reasons, the third button enables or disables manual control. When that button is toggled on, the laser can then be toggled on and off using the fourth button.

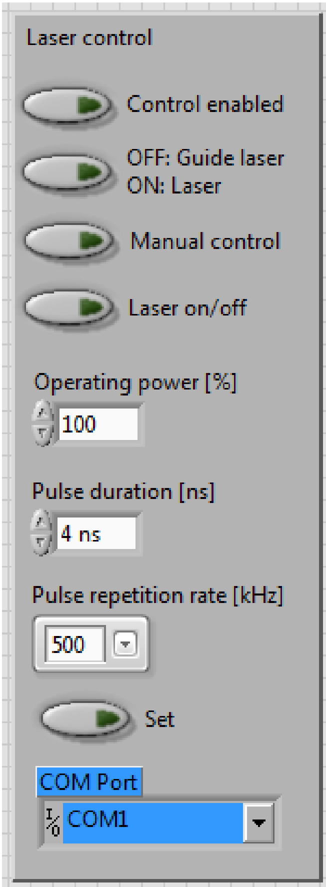


Figure 21. The developed user interface for controlling the laser.

The block diagram for the user interface is seen on figure 22. On the first loop of the program it initializes all the values to false and also initializes the laser by sending the default parameters, which is done using a stacked sequence structure. To send the parameters specified in the interface, there is another stacked sequence structure inside a case structure which is connected to the set button. When it is pressed, the sequence structure sends the parameters one by one to the laser. For the laser control, there is multiple nested case structures and some boolean logic. When the manual control is enabled, the laser is controlled using the button on the panel. When manual control is disabled, the case structure receives the values from global and shared variables, which come from other parts of the program to enable and disable the laser as necessary.

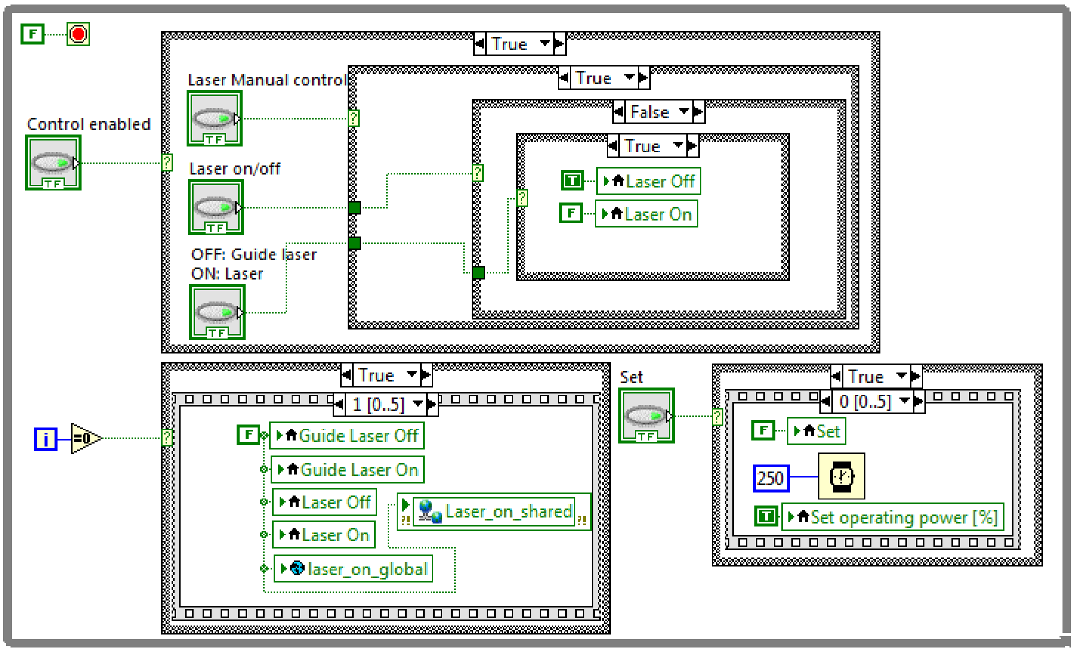


Figure 22. The user interface block diagram implementation.

3.2 Scan head interface

The scan head is controlled using the RTC4 card which is connected to a PCI slot on a desktop computer running LabVIEW. A DLL file is supplied with the interface card which allows external programs to control the scan head using the card's functions. (SCANLAB, 2009)

LabVIEW has a call library -function which allows calling external functions and libraries in the block diagram of a LabVIEW program and also supply function parameters and receive values. This makes it possible to control the scan head using the same LabVIEW program that is used to control the rest of the system. Figure 23 shows an example of such call. The library name, file location and the function name in the library that is to be called is specified. On the parameters page, the values to be passed and the return value are defined. After that, the function appears as a block on the block diagram and has inputs which correspond to the values to be passed to the function and outputs which correspond to the returned values.

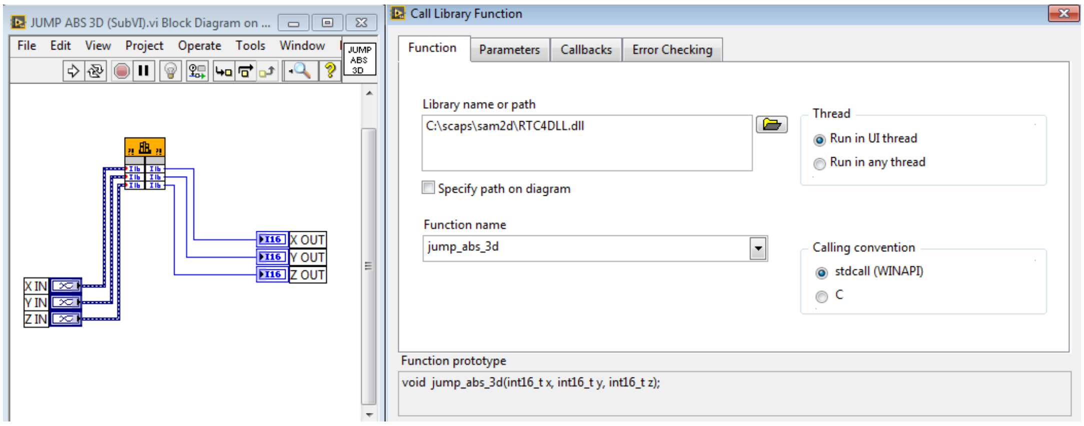


Figure 23. Using external libraries from LabVIEW.

The available functions are specified in the RTC4 manual. They include the initialization commands which must be called every time when starting up the system. There are functions to set and read parameters and settings of the card and the scan head and to monitor the status of the system. In addition, there are the list commands which allow to build a list of specified coordinates to which the laser should move between and then execute the movement. There are also commands to draw arcs and circles directly in addition to lines.

The developed scan head interface program first initializes the scan head when starting up the program. After that it receives commands from other parts of the program and passes them on to the scan head to move the laser to the desired locations to perform scribing. The initialization procedure can be seen in figure 24. First it enables a shared variable which lets the other parts of the program know that the scan head is starting up. After that the required correction file is downloaded as discussed earlier. Then a program file is downloaded which starts the signal processor and resets the scan head. The laser control mode is then set according to the type of the laser. Finally, the jump and mark speeds are set (the movement speed of the laser pointer), the pointer is positioned in the center and the shared variable is disabled meaning that the startup is completed. Since the scan head library functions are asynchronous, the program waits a few hundred milliseconds between the function calls which ensures that the previous step has been completed before the next call.

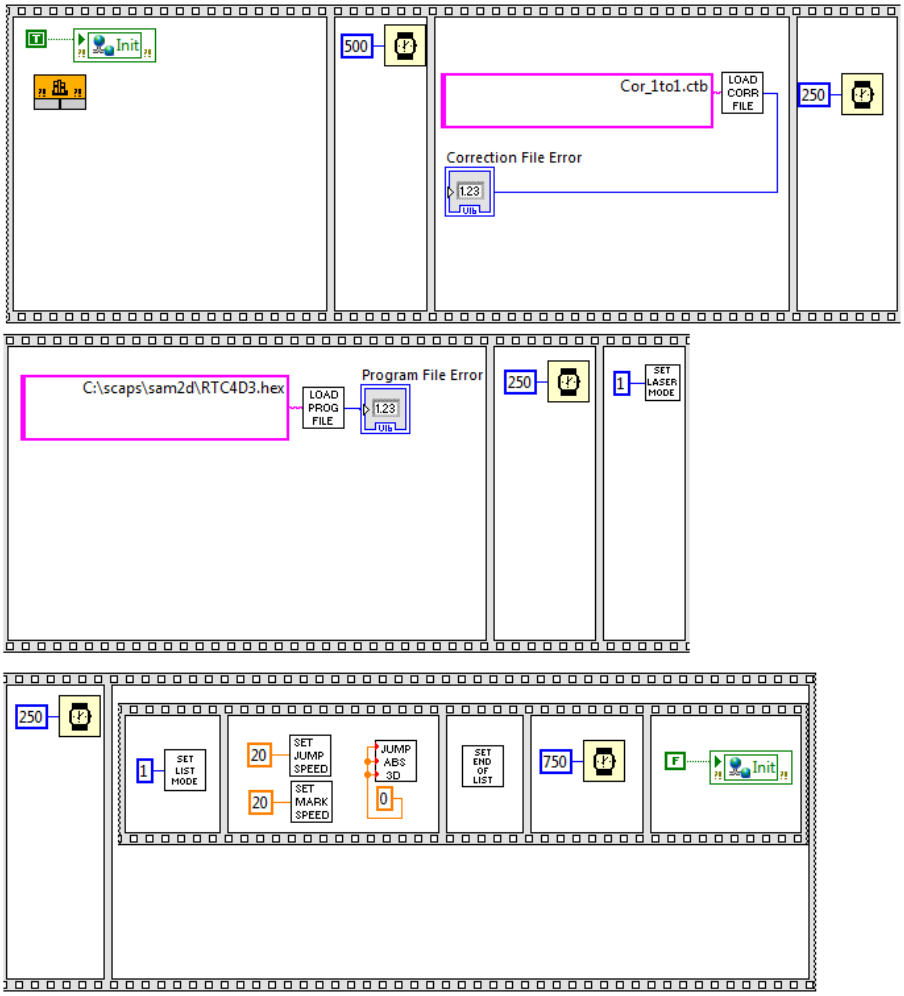


Figure 24. Initialization procedure of the scan head in LabVIEW environment.

3.3 Linear motor control

For controlling the linear motor stage and also the servo motor driving the vertical axis, the SoftMotion -module available in LabVIEW is used as previously explained. It has function blocks available for initialization and setting up the drives, fault handling and many moves such as multi-axial lines, arcs and contouring.

When powering up the drives, a homing run must always be first done to enable accurate positioning. This can be done either in the configuration tool of the drive or in SoftMotion. The process is illustrated in figure 25. The homing run procedure first moves the platform so that it encounters a limit switch at the end of the track. This is marked with a red line in the picture. Then it reverses movement direction and moves very slowly until it encounters the index switch, which is located near the limit switch and illustrated with a green line. When encountering the index switch, the drive stops instantly and zeroes its position measurement. This way the drive will always have the same zero position and will position itself accurately to the same position every run. This run must be performed every time power is lost and for each axis individually.

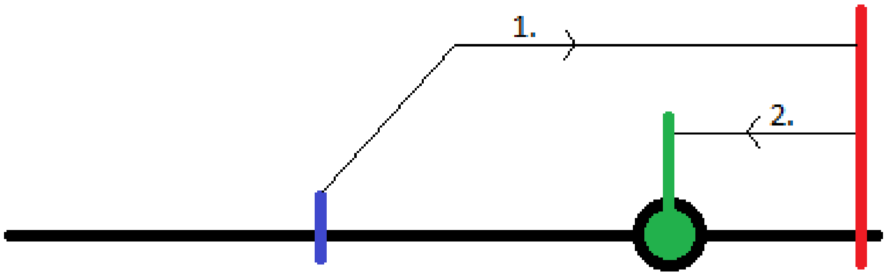


Figure 25. The basic principle of the homing run. Blue line marks the starting point, red line is the end limit and green dot is the home position with the index switch. Arrows and numbers indicate the movement order of the platform.

Figure 26 shows the initializing of the drives in LabVIEW after the homing run is completed and the basic positioning of the drives using SoftMotion. The drives are referred as axes in SoftMotion. Two or more axes can then be combined to a coordinate space and they can be controlled together to perform two- or three-dimensional moves. To power on the drives, a power block can be used. It receives the motion resource from the coordinate space and two boolean values which enable the drive and axes. They go through OR gates so that the drives can be enabled and disabled using the button or via a shared variable from a different part of the program. In case there are faults, the drive operation halts until they are cleared. After fixing the possible issues, this can be done using the Clear faults -block.

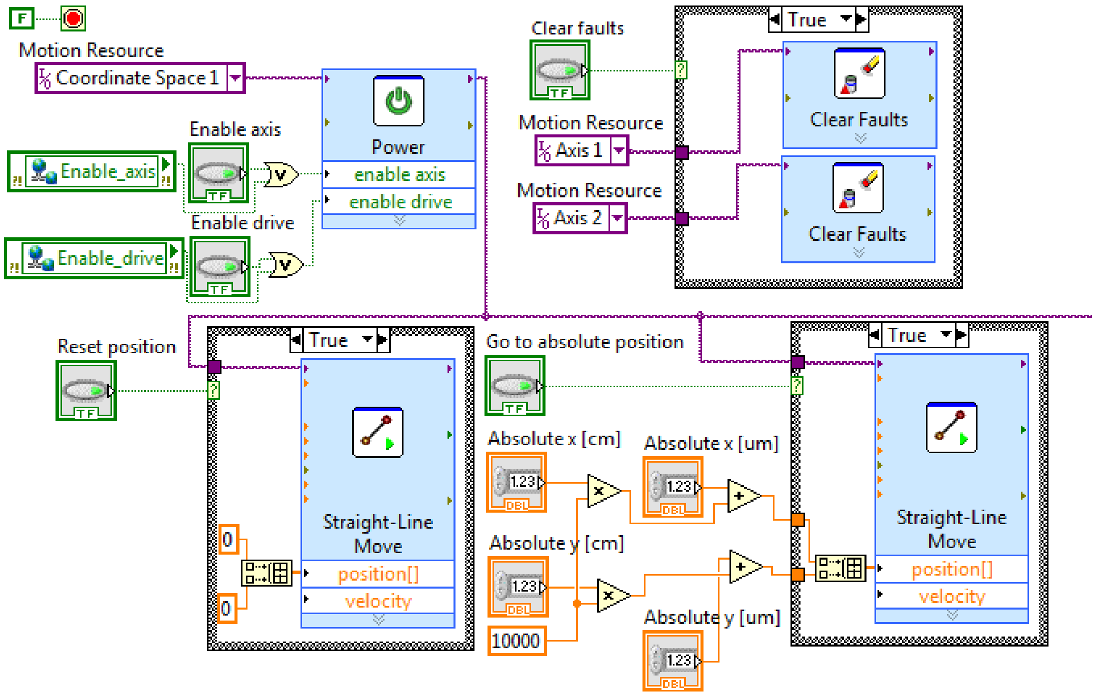


Figure 26. The initialization, startup and basic movement functions of the linear motors.

To reset the position to the zero starting position, a straight-line move with positions set to zero can be used. When using the straight-line move block with a motion resource, both axes positions have to be specified when calling the function and also whether the move is to a relative or absolute position.

To read the current position of the stage, the read -function block can be used. This outputs the absolute positions in micrometers from the home position. If the position is to be displayed in centimeters, the output has to be divided by 10000 as seen in figure 27.

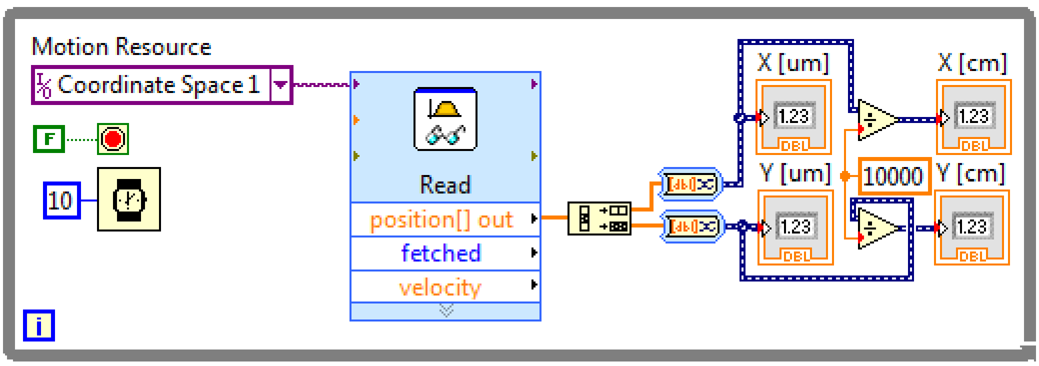


Figure 27. Reading the current position and displaying it to the user.

3.3.1 Vertical movement control

The vertical movement is implemented with a servo motor and exactly the same kind of controller that is used for the linear motors. Therefore, the same straight-line move function block for moving the scan head up and down can be used. The only exception is that the movement is only one-dimensional as compared to the two-dimensional movement of the workpiece.

For the homing run in the vertical axis, two inductive sensors were installed near the other end of the movement area. One serves as the negative limit switch and the other one as the home switch. This is sufficient for positioning since if the homing run is always started to the same direction, the same limit switch is hit every time and the direction is then changed to find the index switch. The positive limit switch can then be implemented programmatically in the servo controller's software.

The motor is actuating a ball screw linear actuator which generates the linear movement. The ball screw actuator has a lead of 5 mm and it is connected through a reduction gear with a gear ratio of 2. In other words, every rotation of the motor shaft equals to 2,5 mm of linear movement.

3.4 Automation and synchronization

Since the linear motor working area is larger than the scan head working area, the linear motor area was divided into 5 separate areas in both directions, which gives 36 separate working areas as the edge can also be used as one area. This is illustrated in figure 28. The linear motor actual working area length is 29,4 cm, but the scan head area fits 5 times inside of it, giving a total area length of 27 cm.

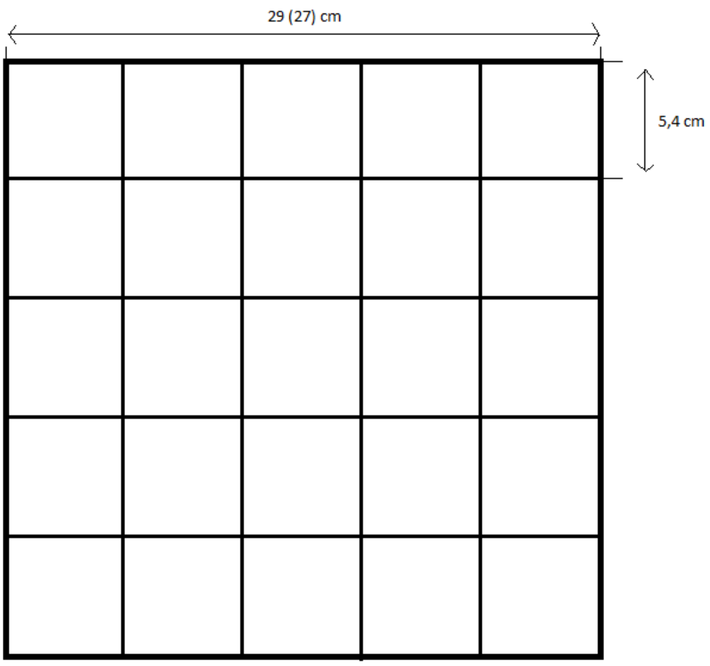


Figure 28. Dividing the larger working area to smaller areas to process with the scan head.

A program was developed to move between the different working areas using manual commands or automated commands from other parts of the program. It is implemented using a state machine. It can either increment or decrement the position by one or go to a specific position with the supplied coordinates. The idle state is shown in figure 29. It receives the commands either from manual pushbuttons or via shared variables from other parts of the program. With boolean logic it then decides which state to go next.

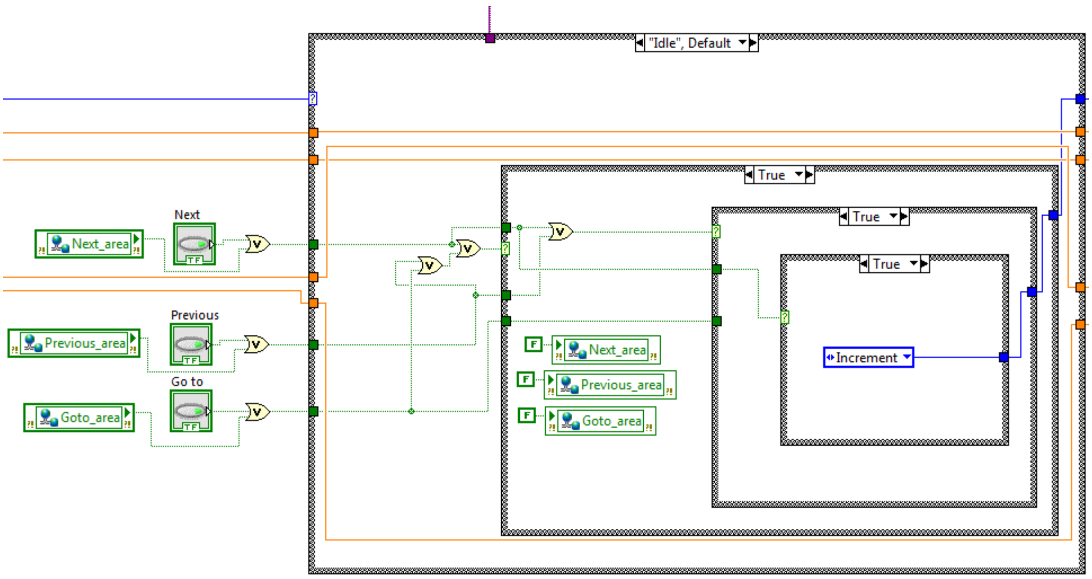


Figure 29. Idle state of the state machine.

In the increment state the program simply increments the variables by one. Logic is built so that if the end of one axis is reached, the program then increments the other axis by one and resets the other axis to zero. This can be seen in figure 30 where the increment state is shown. This way it will go through every position one by one in order. If the end of the area is reached (so both of the axes are at the end limit), the program will reset both of the axes to zero at the next incrementing move. The decrement state is similar but to the opposite direction. After these calculations the program will go to the run state and actually move the platform.

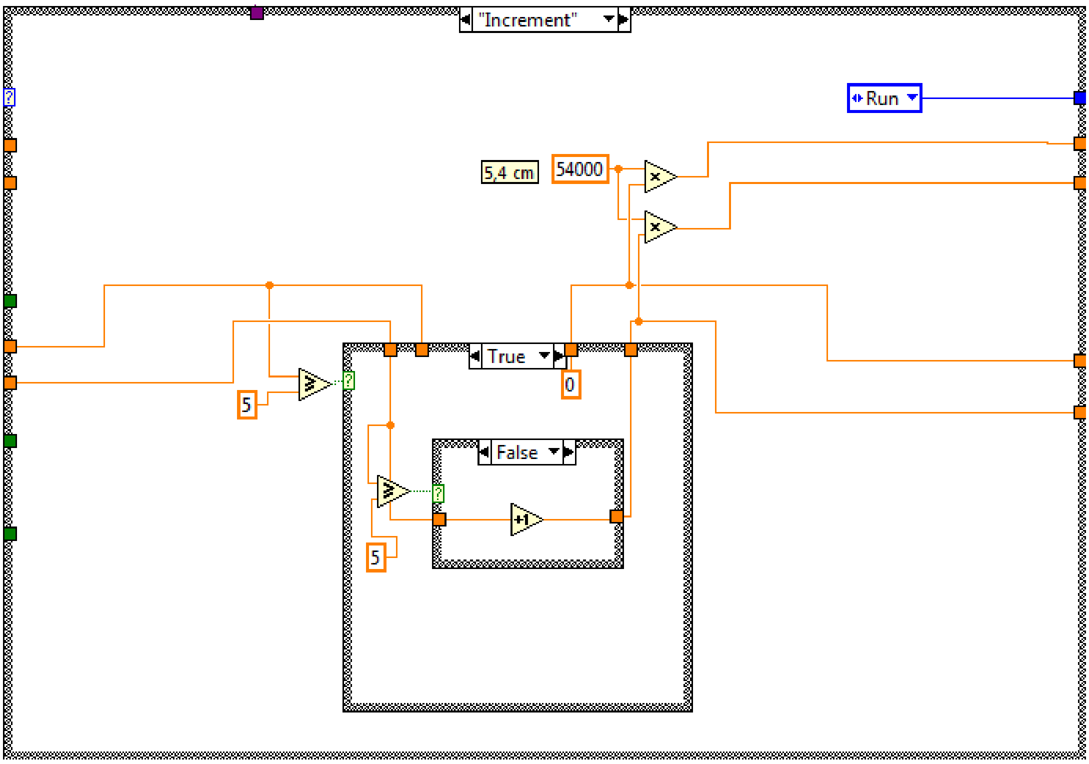


Figure 30. The increment state of the state machine.

The go to -state receives the coordinates via shared variables, updates them to the state machine and then simply moves to the run state to execute the movements. This is shown in figure 31.

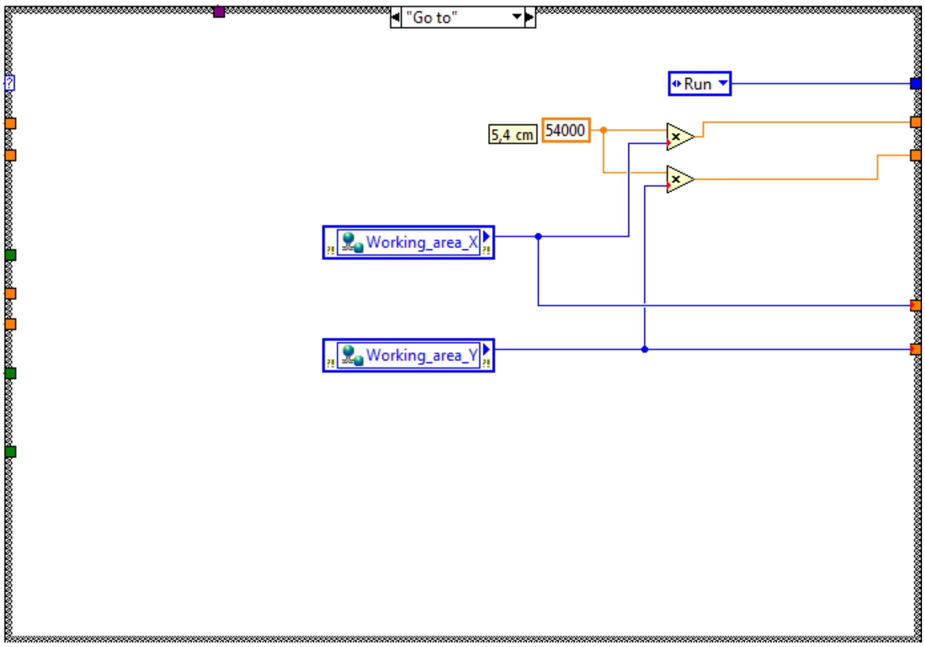


Figure 31. The go to -state of the state machine.

Finally, in the run state shown in figure 32 the program gets the position from the previous state as explained and moves the stage to the new position with the set velocity. It also updates the new coordinates to the local variables so they can be displayed in the graphical interface. After this the state machine resets to the idle state again.

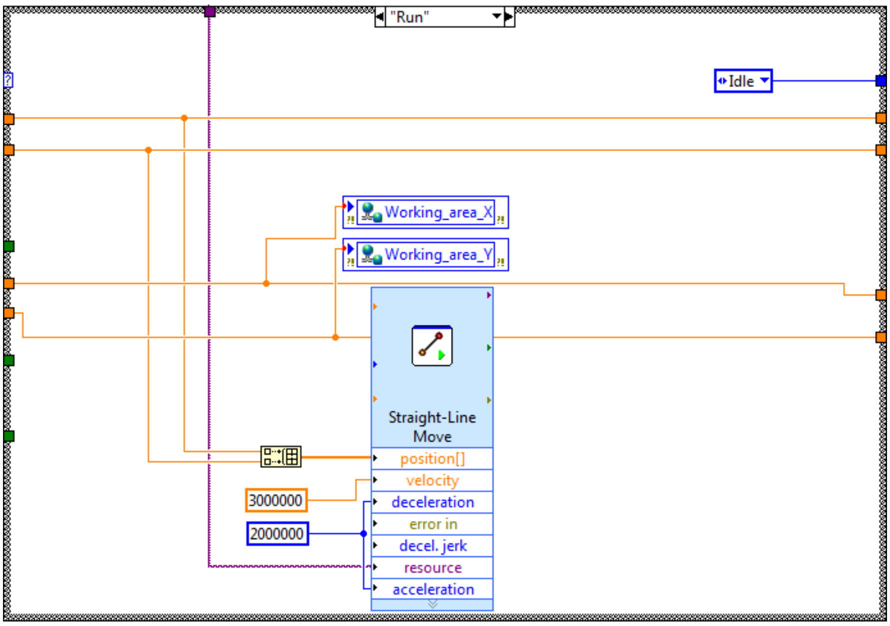


Figure 32. The run state of the state machine.

Synchronous function is a function which blocks the program execution while it is running. In other words, it makes the rest of the loop execution hang while waiting the function to complete before continuing. In contrast to this are asynchronous functions, which do not block the execution of the program when called. With synchronous functions it is easy to manage the timing and execution order of the program, since the rest of the loop waits for the function to complete. With asynchronous functions it is necessary to create separate logic beside the functions which control the execution order of the program.

The SoftMotion functions can be chosen to be either synchronous or asynchronous. The program flow is easier to control with synchronous functions, so they are used in this program. However, the functions for the scan head are only available as asynchronous. Therefore it is necessary to come up with a different way to manage the execution and timing of the program when performing scribing with the scan head.

To accomplish the scribing of different kind of shapes, LabVIEW has a drawing library that is used to draw the shapes and lines to a drawing area. The coordinates for these shapes are acquired from mouse movements and button presses, stored in arrays and sent to the part of the program which actually performs the movement of the scan head and linear motors.

One way to perform the scribing is using the linear motor stage. It is able to perform scribes so that the laser is stationary. This means that the scan head is not used but the laser beam is simply passing through it and the linear motors are used to move the workpiece around so that the laser scribes the shapes. This makes it possible to use the entire working area and process larger workpieces as a complete area. The program is implemented using a state machine and network variables that communicate with other parts of the system.

When the state machine receives the command to begin, it first goes to increment state as seen on figure 33. This state increments the loop counter by one and prepares the next move by reading the next X and Y coordinates from the array. It also checks if the end of the array has been reached. In that case, it goes back to the idle state. In addition, this state reads whether the laser should be on or off for the next move, and the “color” of the next move in the graphical interface. This means the vertical position of the scan head, since with different focus heights it is possible to produce different colors by using a special method. These arrays are constructed when the shapes are drawn onto the drawing area in the user interface.

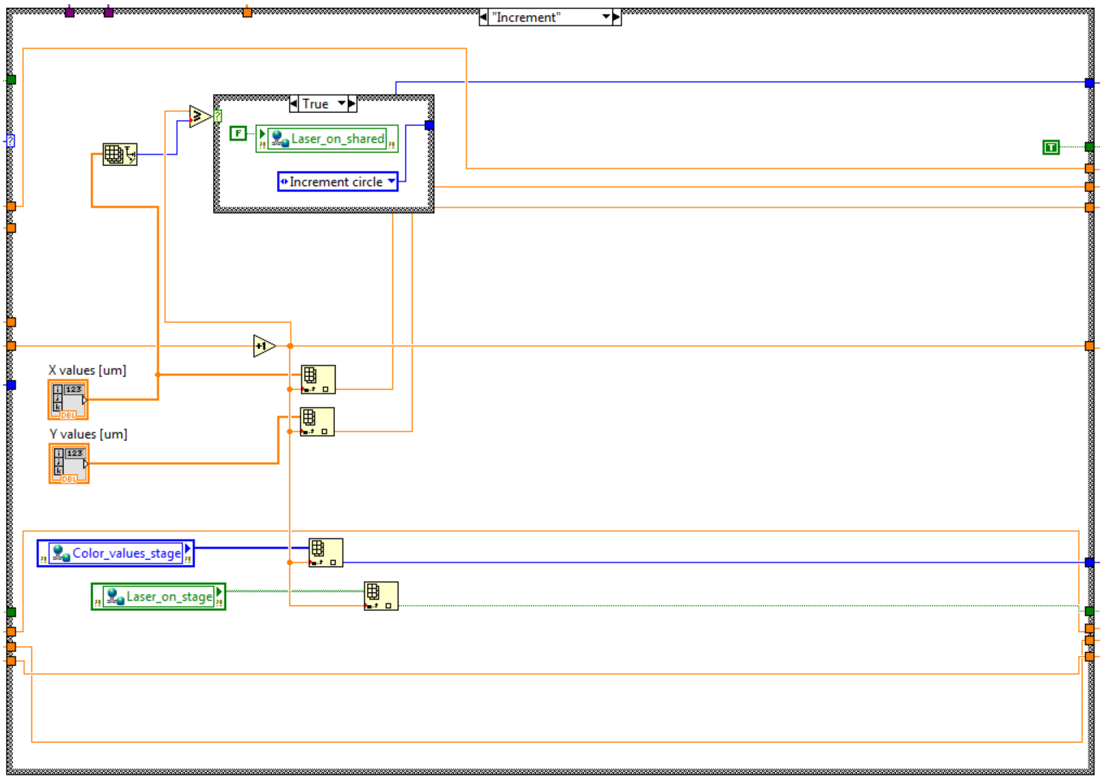


Figure 33. Increment state of the linear motor state machine.

After the increment state the program goes to the run state, which executes the movements. First the program checks if the laser is supposed to be on for this move. If not, it will move the linear motor stage and the vertical axis simultaneously to the next position and keep the laser turned off. This is seen on figure 34. The linear motor stage moves faster if the laser is off to make the processing faster.

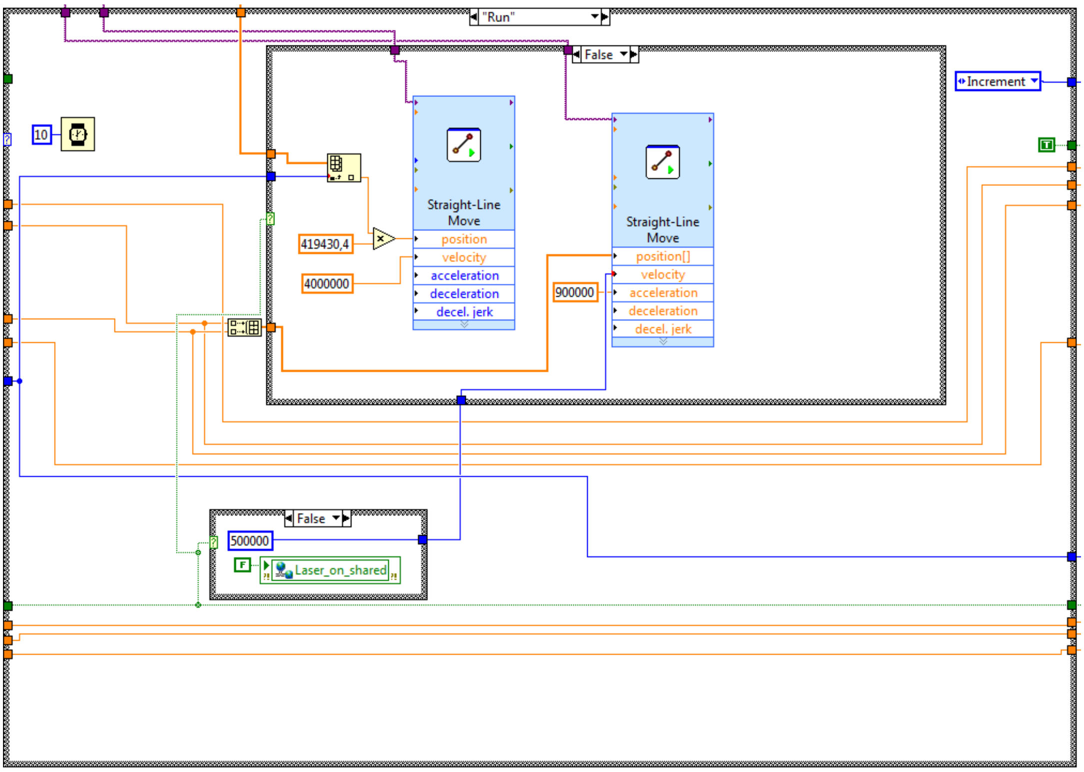


Figure 34. Run state of the state machine when the laser is not needed to be turned on.

However if the laser is required to be on for the move as seen on figure 35, the procedure is more complicated. Now the scan head is first moved to the correct position, then the laser is turned on and at the same time the movement of the platform is started. On LabVIEW the gray dotted line box is a flat sequence structure and it means that the program will execute in order from left to right as previously explained. After this, the laser is turned off and the program goes back to the increment state to prepare for the next move.

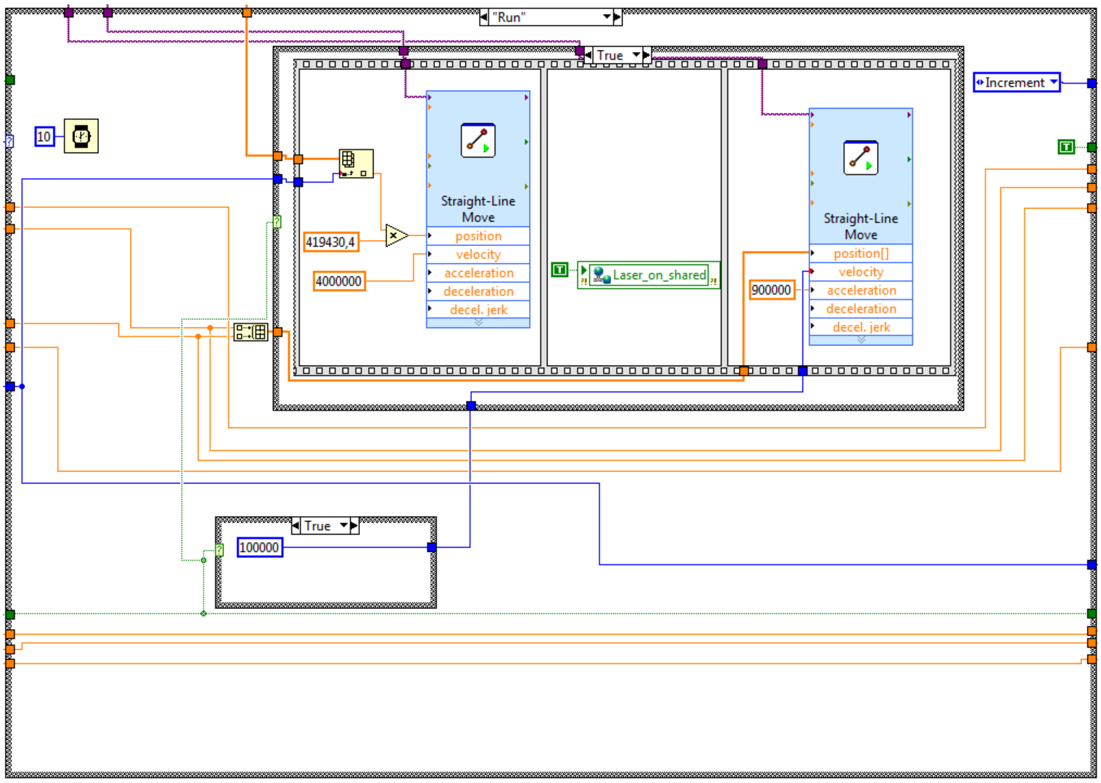


Figure 35. Run state of the linear motor state machine.

It is also possible to perform circles in addition to other shapes. For this, SoftMotion has an arc function that moves the platform in a circular motion. These are scribed after other shapes. First the increment state reads the position of the circle, its radius and the position of the scan head from the arrays like previously described. This is seen on figure 36.

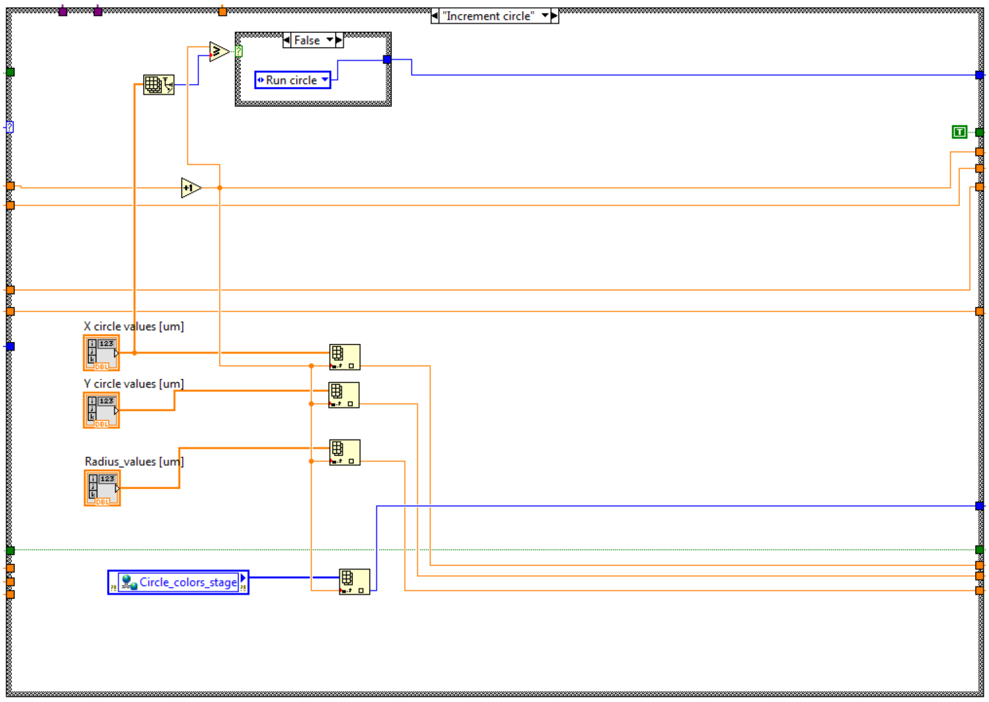


Figure 36. Increment state for drawing circles with the linear motor.

In the run state, the platform and the vertical axis are first moved to the position where the circle is to be drawn. Then the laser is turned on and the arc move executed as seen on figure 37. Finally, the laser is turned off and the state machine goes back to the increment state.

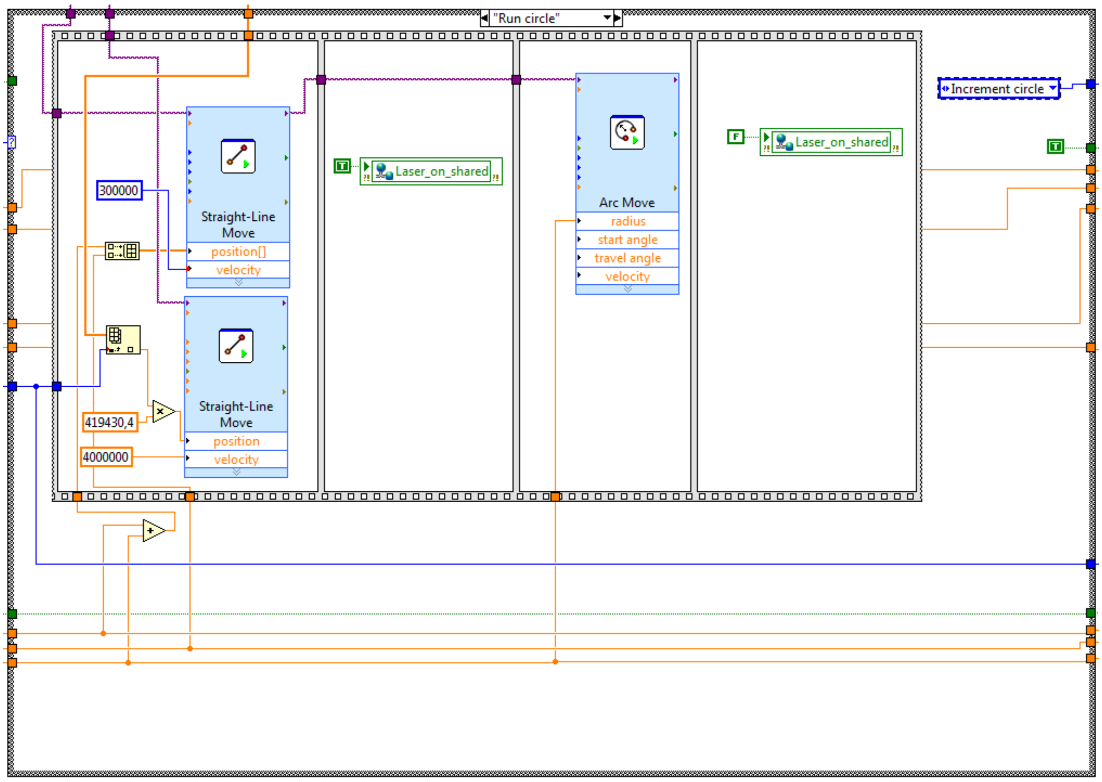


Figure 37. Run state for drawing circles with the linear motor.

After all the scribes have been completed, the state machine goes to the finished state which is seen on figure 38. The platform and the scan head is moved to zero position and it is made sure that the laser is turned off. Then the program goes back to the idle state to wait for the next command.

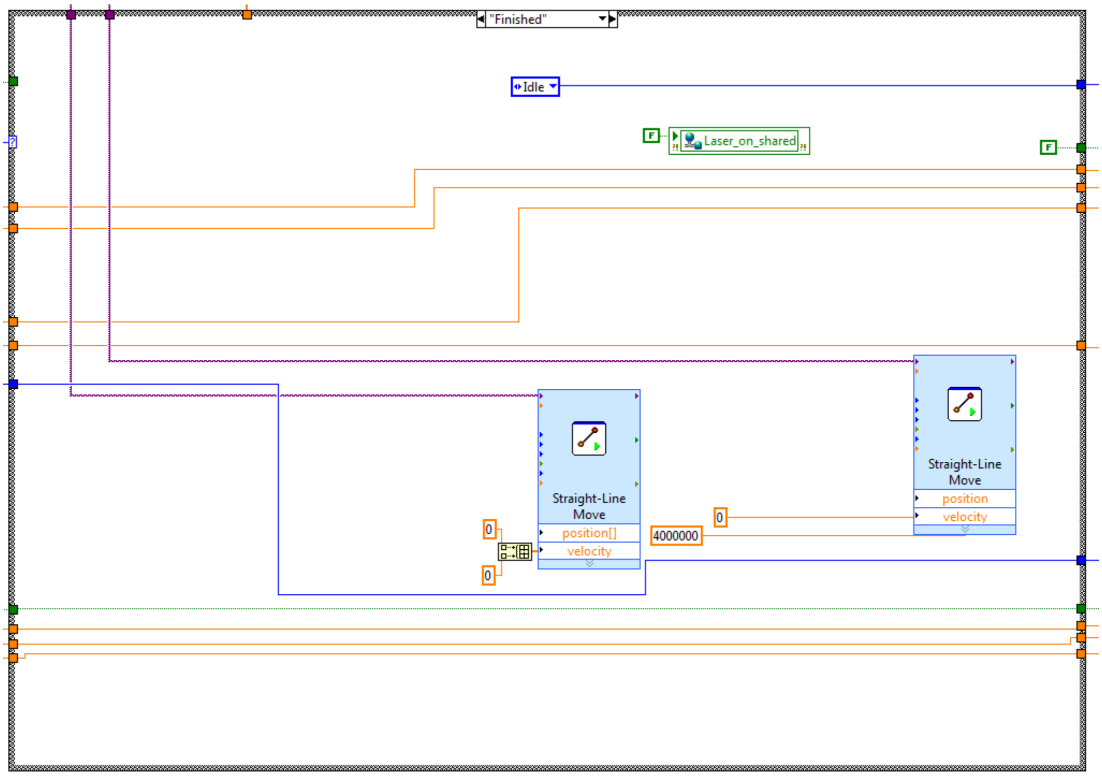


Figure 38. Finished state.

To perform scribing with the scan head so that the linear motor platform is stationary and only the small scan head area is used, the asynchronous functions of the RTC4 card library need to be used. It is accomplished by using a partly similar state machine as seen previously and adding some more logic to control the execution and timing, since the functions are asynchronous.

Figure 39 shows the increment state of this state machine. It acquires the next X and Y coordinates of the platform, the next vertical position of the scan head and the next state of the laser from the arrays. Since the graphical interface drawing area has 540 pixels, the values are divided by that value to get a scalable value of the position between zero and one. The scan head has 65 535 possible laser positions, so the scalable value is then multiplied by it. The accuracy of the scribing can also easily be increased by simply increasing the size of the drawing area. Moreover, since the zero position is at the center of the area and the coordinates should start from one side of the scribable area, half of the length of the area if subtracted from the value.

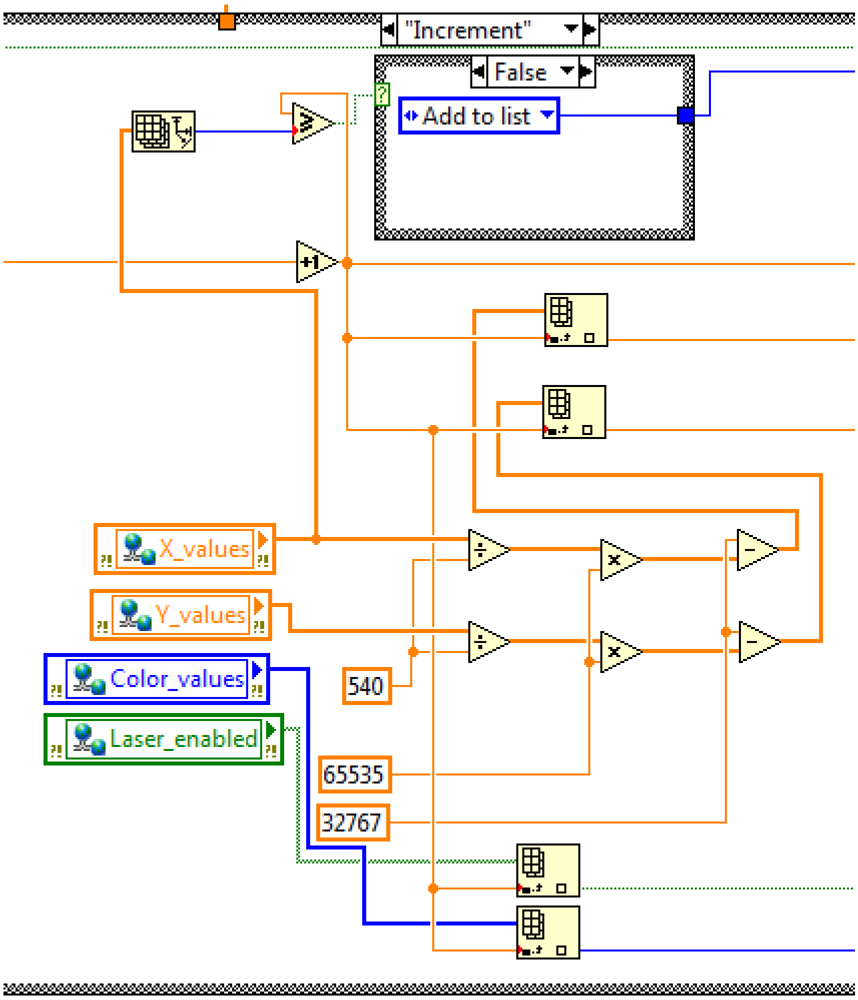


Figure 39. Increment state of the scan head program.

After that, the program goes to the run state as previously. First it is checked if the vertical position has changed from previous iteration of the loop. If that is the case, the laser is turned off and by setting the shared variable to true, a command is set to the motor control program which moves the scan head to the new position. Since the functions are asynchronous, there is a while loop in which the program execution gets stuck until the vertical movement is completed. Then the laser is turned on and at the same time the scan head scribing started. Again, there has to be a while loop after that as seen on figure 40 which waits for the scribe to finish before moving on to the next step.

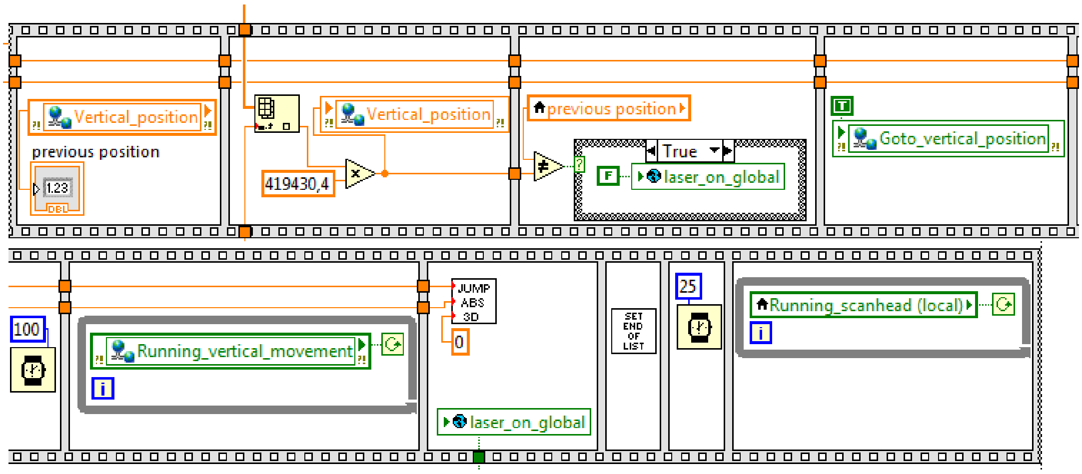


Figure 40. Run state of the scan head program.

To draw circles and arcs, the scan head first positions the laser for where the circle is to be drawn. Then the scan head is moved vertically to the correct position as described previously. After that, the circle or arc draw is started with the arc command and the laser is turned on. This procedure is shown in figure 41.

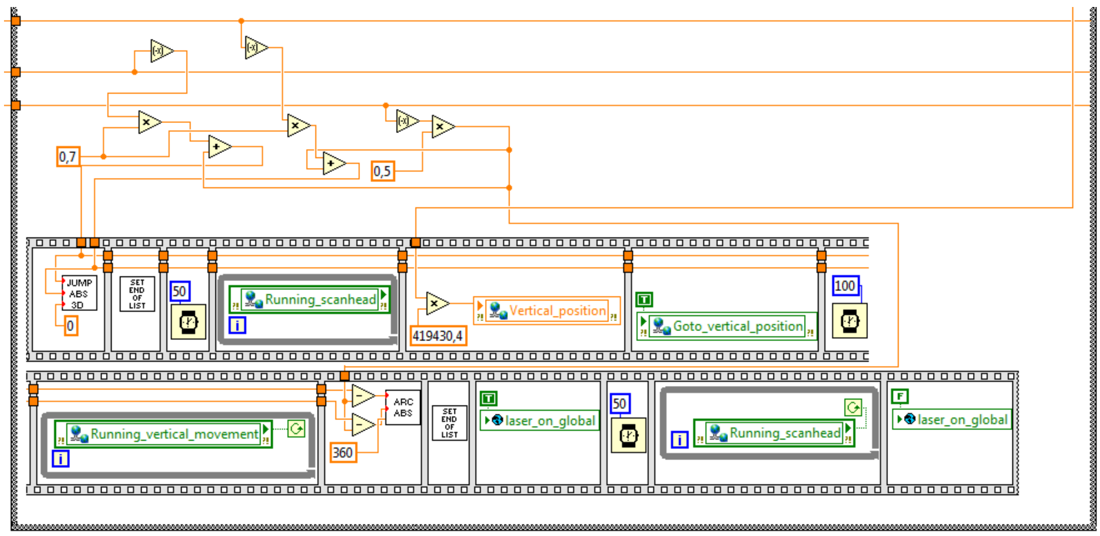


Figure 41. The state for drawing circles with the scan head.

The movement of the linear motor and the scan head can also be combined so that all of the working areas are scribed after each other automatically. For this there is a state machine which calls the other previously explained state machines in order. The execution order sequence structure is shown in figure 42. First the working area is selected and a command is sent remotely to move the working area to the new position. After the movement is finished, the selected working area is scribed using the scan head by sending another remote command to run the scan head program. When the scribe is finished, the next working area is selected and the sequence starts over.

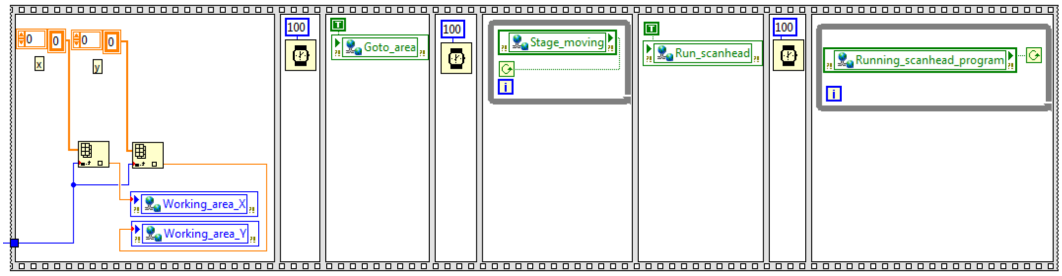


Figure 42. Performing all the scribes of different areas automatically in order.

3.5 Final system

The main graphical interface of the finished program is shown in figure 43. It has the motor and laser controls as explained previously, working area controls and the drawing areas and controls for the scan head and for the linear motor stage separately.

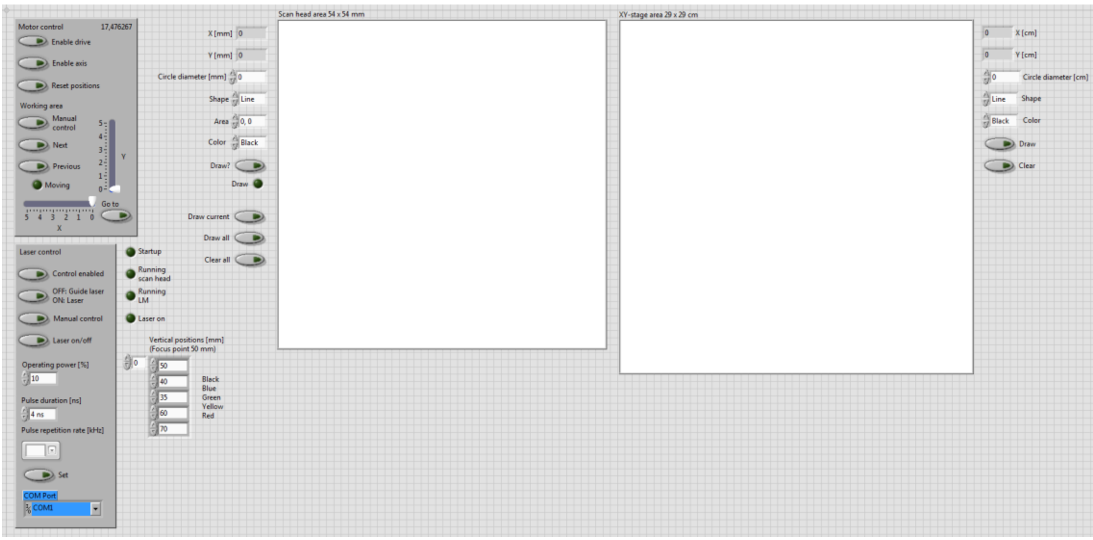


Figure 43. Main user interface of the LabVIEW program.

Physically the laser processing system can be seen in figure 44. The linear motor stage is positioned under the scan head and fastened on to an optical table. The scan head is attached to the ball screw actuator and the laser is entering it via the yellow optical fiber and the laser output head.

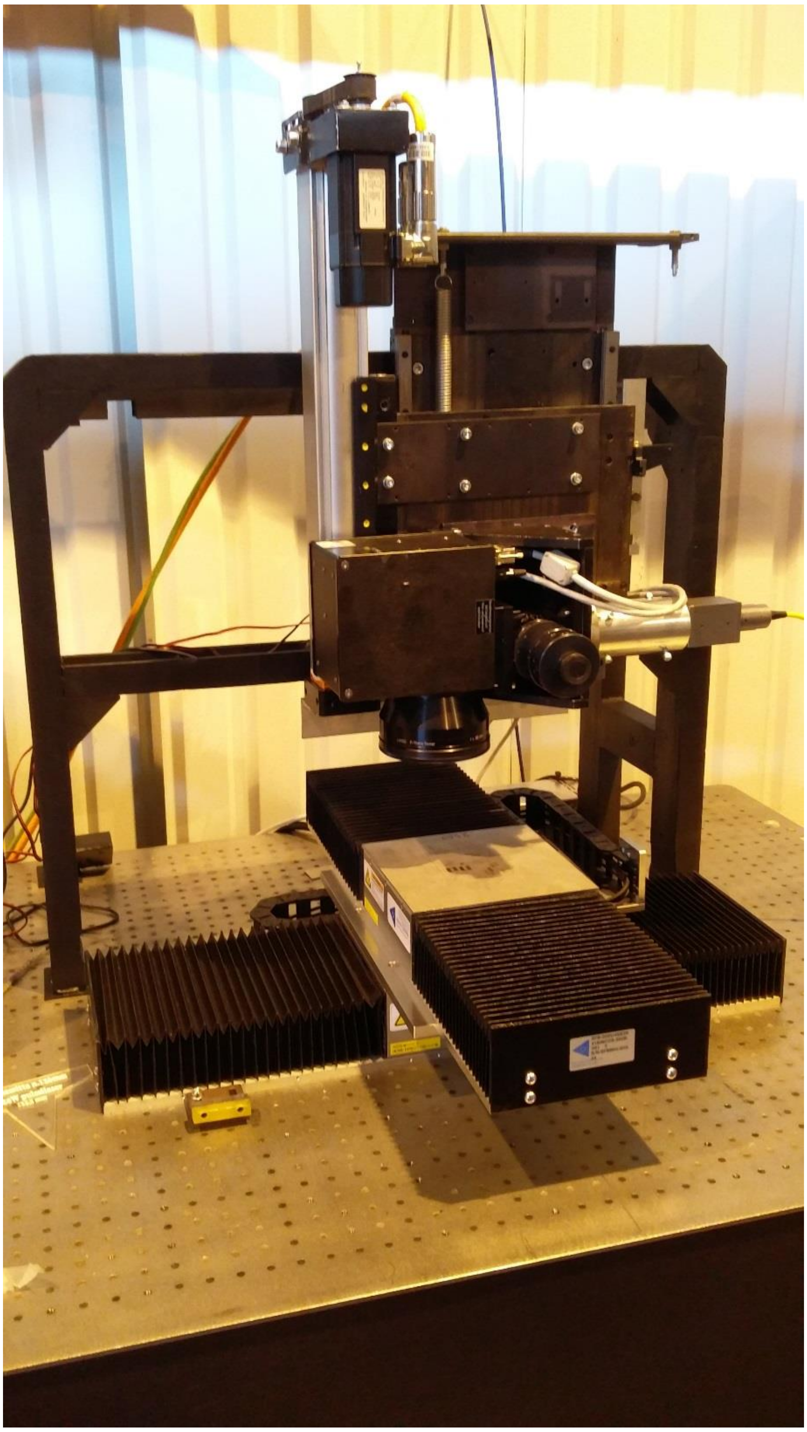


Figure 44. The finished laser processing system.

A steel plate to protect the linear motor from the laser and to hold the workpiece in place was machined. This can be seen in figure 45. On the plate is also attached an example workpiece which can be scribed using the system.

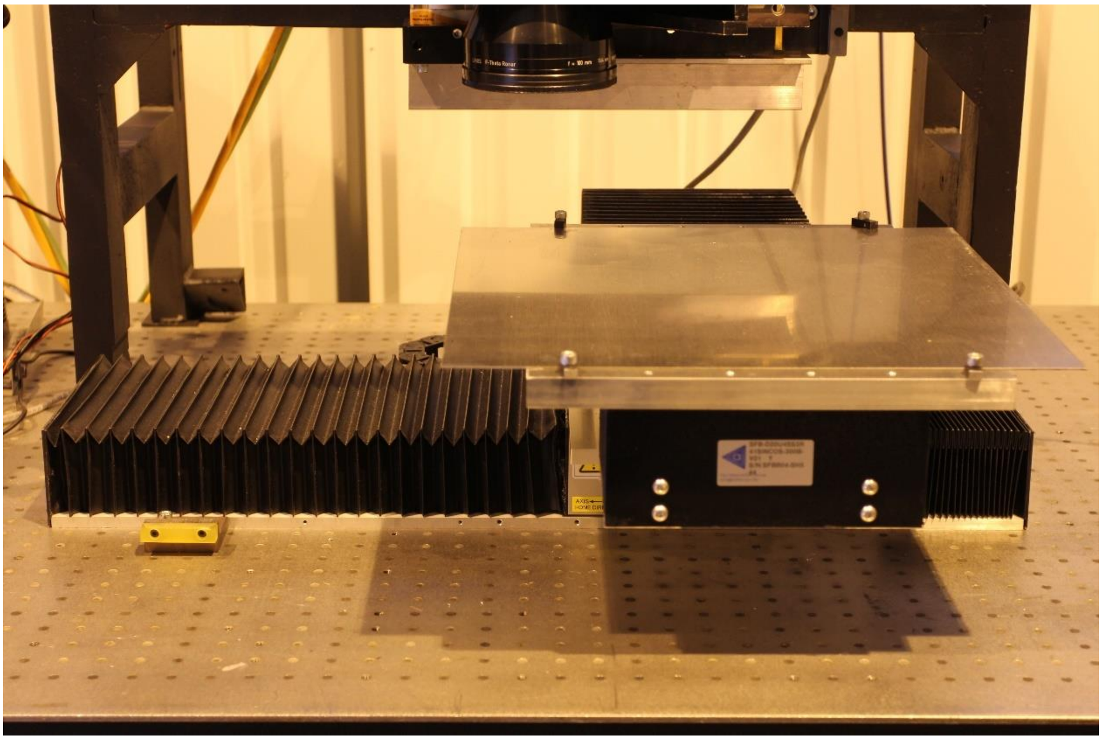


Figure 45. Close-up of the linear motor system and the platform for the workpiece.

3.5.1 Color marking tests

Different colors can be generated by laser pulses on metal surfaces, as explained previously. Some tests were conducted at the end of this thesis to find color variations that can be generated with the system. To achieve better results, more distinguished colors and different shades, further research will be done in the future. Table 5 shows some parameters of the laser and the difference of the scan head height from the focus point that generate different colors.

Table 5. Some possible combinations of laser parameters and scan head height which generate specific colors on a stainless steel plate.

Color	**Laser power [W]**	**Pulse duration [ns]**	**Repetition rate [kHz]**	**Difference from focus height [mm]**	**Line gap [mm]**	**Scribe speed [mm/s]**
Red	10	30	40	0	0,05	1000
Brown	10	100	40	0	0,01	1000
Black	10	14	40	1	0,01	1000
Silver	10	100	500	0	0,01	1500
Orange	20	14	40	-1	0,01	1000

The testing was conducted by doing samples of 12 squares and varying the parameters between them systematically. Figure 46 shows part of this testing process.

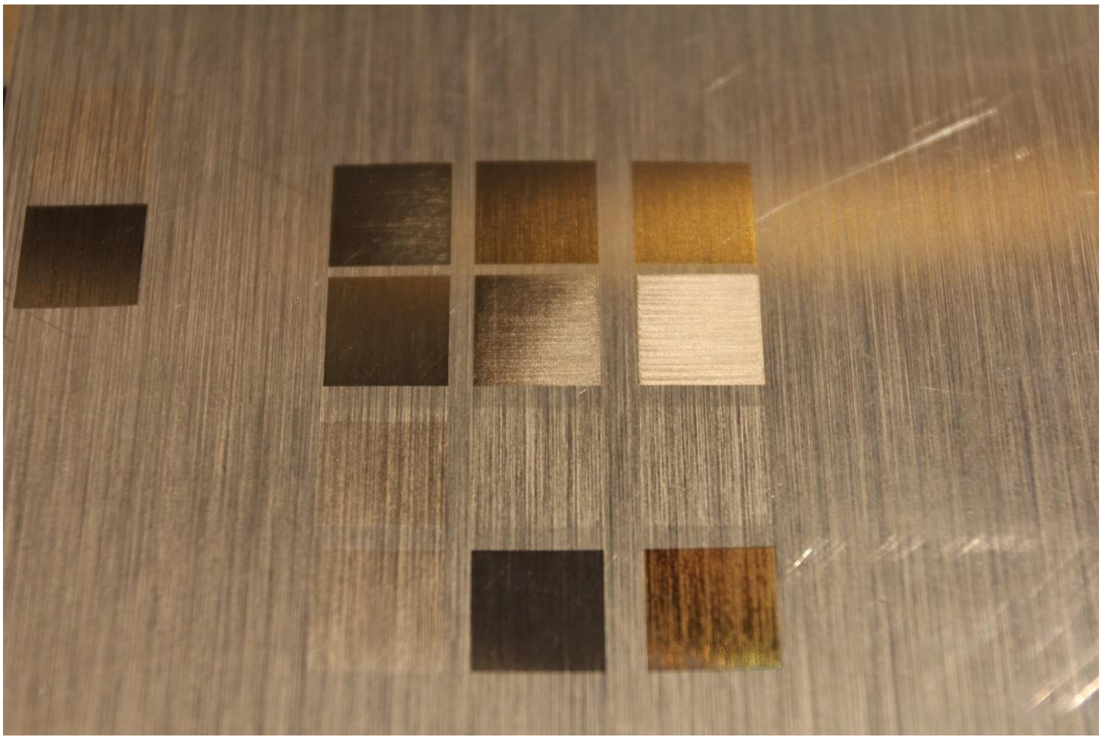


Figure 46. Testing of the color generation by varying the parameters of the laser and height of the scan head.

The most distinguished colors and uniform quality samples are shown in figure 47. They are created with the parameters shown on table 5 to a stainless steel workpiece. In the picture some of them look like they are the same color, but the shade varies greatly depending on the viewing angle as previously explained.

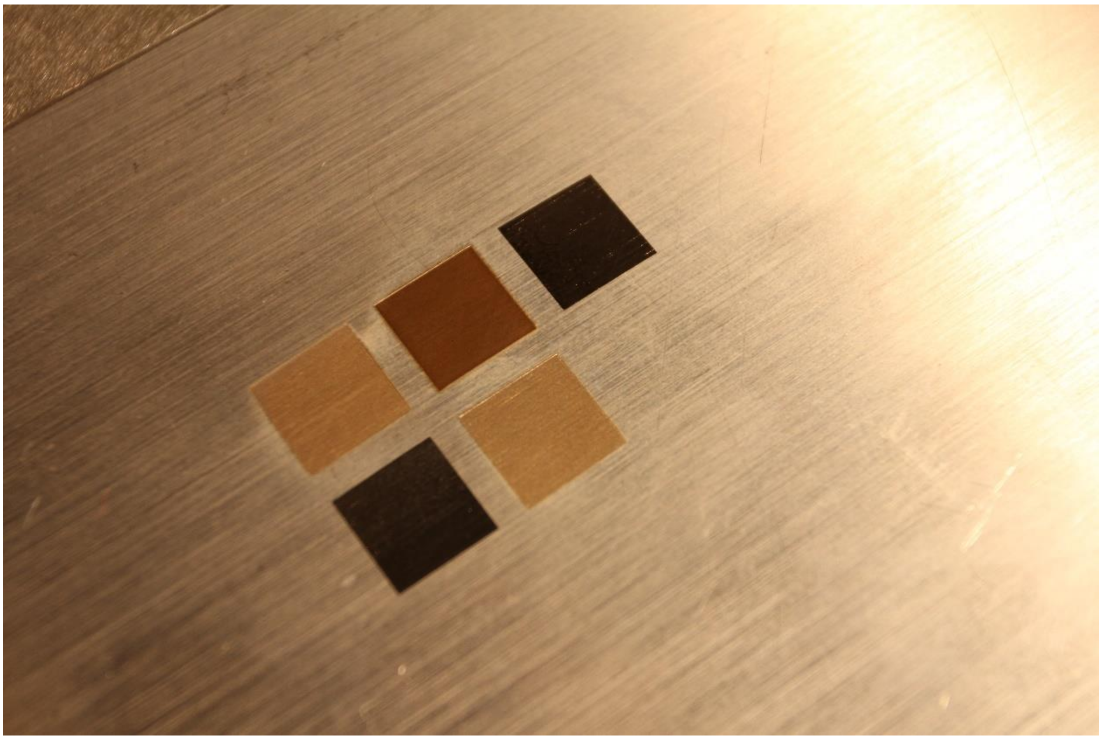


Figure 47. The best distinguished samples generated by varying the parameters. The actual colors look different and depend greatly of the angle of the incoming light and the viewing angle.

4 CONCLUSIONS

The goal of this thesis was to first select and order a suitable motion system for moving the workpiece in a two-dimensional area and for moving the scan head vertically up and down. The main goals after that were to develop a program which combines all of the parts of the laser processing system into one, allows them to be controlled from the same interface and makes it possible to perform scribing using the scan head, linear motor and both at the same time.

The motors that were selected are well suited for the tasks necessary. The linear motor and servo motor motion control and communication works excellently and makes almost any kind of movement possible. Nevertheless, there were some issues with the scan head communication. The scan head library has a mark function, which would allow better scribing than the current setup. However, this function was not communicating with the laser properly for some reason and it could not be made to work.

In addition, the drawing area for scribes of the LabVIEW program could be more sophisticated for example to allow copy pasting of shapes from images or to allow entering text or saving the work after shutting down the program. Having said that, it is beyond the scope of this thesis to develop a more complicated program since it would take a lot of time. The current program still allows drawing of many kinds of shapes using either the scan head or the linear motor, the scribing of solar cells and laser color marking for example. It also allows communicating with the laser and other parts of the system from the same interface.

LabVIEW as a development platform for this application is also a good choice since it makes it easy to communicate and use many different protocols and libraries from the same program, such as EtherCAT fieldbus, serial port communication, DLL libraries and so on. However, for the actual scribing work, the LabVIEW drawing library could be more sophisticated to make it easier to perform different processing tasks.

REFERENCES

Aerotech, Inc. 2010. *Linear motors application guide*. Brochure.

Aerotech, Inc. *Resolution, Accuracy and Repeatability.* Engineering reference.

All on Robots. *Cartesian robots.* [Online]. Available from:
http://www.allonrobots.com/cartesian-robots.html
[Accessed 01.05.2016]

Bosch Rexroth. 2006. *Linear Motion Technology Handbook.*

Bovatsek, J., Tamhankar, A., Patel, R. S., Bulgakova, N. M. & Bonse, J. 2010. *Thin film removal mechanisms in ns-laser processing of PV materials*. Thin Solid Films, 518. Pages 2897-2904.

Brumson, B. 2001. *Scara vs. Cartesian Robots: Selecting the Right Type for Your Applications*. [Online]. Available from:
http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Scara-vs-Cartesian-Robots-Selecting-the-Right-Type-for-Your-Applications/content_id/1001
[Accessed 01.05.2016]

Energy.gov. *Copper Indium Gallium Diselenide.* [Online]. Available from:
http://energy.gov/eere/sunshot/copper-indium-gallium-diselenide
[Accessed 21.05.2016]

EtherCAT Technology Group. *Technical Introduction and Overview.* [Online]. Available from:
https://www.ethercat.org/en/technology.html
[Accessed 08.05.2016]

Festo. *Technical Support Wiki, Linear motors.* [Online]. Available from:
https://www.festo.com/wiki/en/Linear_motors
[Accesssed 16.04.2016]

Gieras, J., Piech, Z., Tomczuk, B. 2011. *Linear Synchronous Motors.* 2nd ed. Connecticut. CRC Press. ISBN 978-1-4398-4221-8.

Hagl, R., Heidenhain, J. 2000. *Encoders keep linear motors on track*. Machine Design. [Online]. Available from:
http://machinedesign.com/technologies/encoders-keep-linear-motors-track
[Accessed 31.04.2016]

Hyatt, G. *Ballscrews or linear motors?* [Online]. Available from:
https://www.makino.com/about/news/Ballscrews-or-Linear-Motors/124/
[Accessed 30.06.2016]

Hyun, D.-S., Jung, I.-S., Shim, J.-H., Yoon, S.-B. 1999. *Analysis of Forces in a Short Primary Type Permanent Magnet Linear Synchronous Motor*. IEEE Transaction on Energy Conversion, Vol. 14, No. 4. pages 1265-1270.

IPG Photonics. 2009. *YLPM Series Pulsed Ytterbium Fiber Lasers User's Guide.*

Kollmorgen. *AKD Servo Drives.* [Online]. Available from:
http://www.kollmorgen.com/en-gb/products/drives/servo/akd/akd-servo-drive/
[Accessed 09.06.2016]

Kollmorgen. 2014. *AKD User Guide.* Revision N.

Laakso, P., Ruotsalainen, S., Pantsar, H., Penttilä, R. 2009. *Relation of laser parameters in color marking of stainless steel.* VTT, Lappeenranta.

Lindh, T. 2014. BL40A2800 *Electrical Motion Control Systems* lecture slides, Lappeenranta University of Technology.

Litmanen, E. 2015. *Laser scribing of thin-film PV modules.* Lappeenranta University of Technology, LUT School of Energy Systems. Bachelor's Thesis. 30 pages.

Moore Industries. 2006. *Introduction to Fieldbus.* [Online]. Available from:
http://www.automation.com/pdf_articles/fieldbus.pdf
[Accessed 07.05.2016]

National Instruments. 2004. *Getting Started with NI SoftMotion Development Module for LabVIEW.*

National Instruments. 2011. *Serial Instrument Control Tutorial.* [Online]. Available from:
http://www.ni.com/tutorial/2897/en/
[Accessed 20.07.2016]

National Instruments. 2013. *National Instruments and Kollmorgen Collaborate to Provide Advanced Servo Solutions.* [Online]. Available from:
http://www.ni.com/newsroom/release/ni-and-kollmorgen-collaborate-to-provide-advanced-servo-solutions/en/
[Accessed 08.06.2016]

National Instruments. *LabVIEW System Design Software.* [Online]. Available from:
http://www.ni.com/labview/
[Accessed 23.06.2016]

NREL. *Solar Photovoltaic Technology Basics.* [Online]. Available from:
http://www.nrel.gov/learning/re_photovoltaics.html
[Accessed 20.05.2016]

Ponomarev, P. 2009. *Control of permanent magnet linear synchronous motor in motion control applications*. Lappeenranta University of Technology, Faculty of Technology, Department of Electrical Engineering. Master's Thesis. 64 pages.

PVEducation. *Solar Cell Operation.* Honsberg, C., Bowden, S. [Online]. Available from:
http://www.pveducation.org/pvcdrom/solar-cell-operation
[Accessed 20.05.2016]

Real Time Automation. *EtherCAT Protocol Overview.* [Online]. Available from:
http://www.rtaautomation.com/technologies/ethercat/
[Accessed 08.07.2016]

Rockwell Automation. *Comparison of Linear Motor and Rotary Servo Motor Systems from Anorad*. [Online]. Available from:
http://www.rockwellautomation.com/global/anorad/products/linear-motors/comparison.page
[Accessed 16.05.2016]

Roozbahani, H. 2015. BK60A1000 *Control of Mechatronic Machines* lecture slides, Lappeenranta University of Technology.

Roozbahani, H. 2015. BK60A1200 *Programming in Control and Mechatronics* lecture slides, Lappeenranta University of Technology.

SCANLAB. 2009. *Installation and Operation, The RTC4 PC Interface Board.* Revision 1.4 e.

SCANLAB. 2011. *Installation and Operation, hurrySCAN II 14.* Revision 2.6 e.

SCANLAB. *RTC Software.* [Online]. Available from:
http://www.scanlab.de/en/products/control-electronics/rtc-software
[Accessed 08.07.2016]

Sigmakoki. *Laser processing systems guide.* [Online]. Available from:
https://www.global-optosigma.com/en_jp/application/guide-laserpro1.html
[Accessed 28.07.2016]

Solar Tech USA. 2014. *Types of Solar Cells.* [Online]. Available from:
http://www.solartech-usa.com/types-of-solar-cells
[Accessed 21.05.2016]

Tecnotion. 2008. *A primer of Technotion Linear Motors.* Version 2.1. Document number 4022.363.4187.2. [Online]. Available from:
http://www.grp6.com/documents/Basic_Tutorial_Information_Linear_Motors.pdf
[Accessed 20.05.2016]

Vaughn, R. 2013. *The Difference between Cartesian, Six-Axis, and SCARA Robots*. Machine Design. [Online]. Available from:
http://machinedesign.com/motion-control/difference-between-cartesian-six-axis-and-scara-robots
[Accessed 01.05.2016]

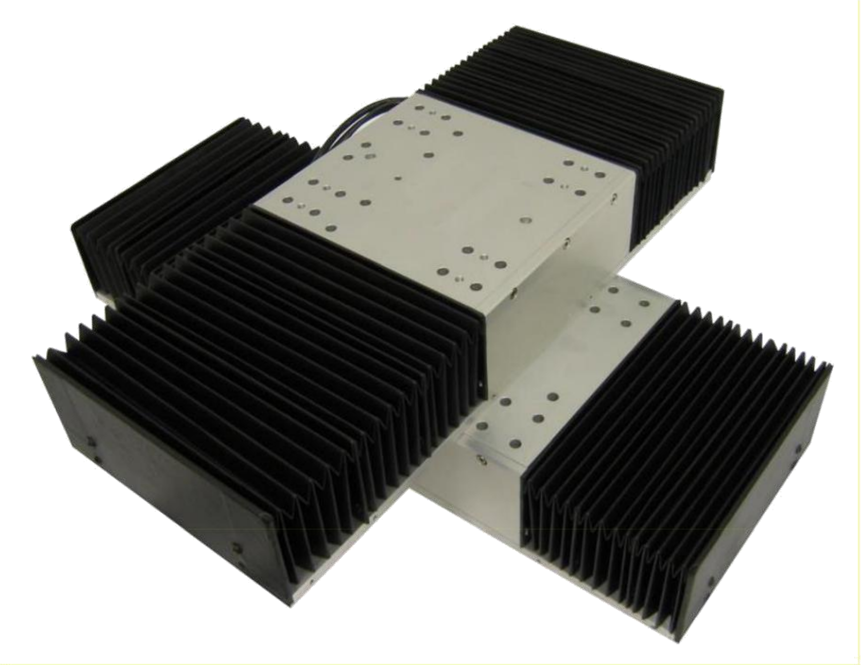
Xu, I. 2016. *Stages and Systems*, Akribis systems manual.

Zvyagina, D. 2010. *Applications and markets of linear motors*. Lappeenranta University of Technology, Faculty of Technology, Department of Electrical Engineering. Master’s Thesis. 80 pages.

Multiaxial linear motor stage technical parameters

DGB Series

XY Dual Guide Stage with bellow cover

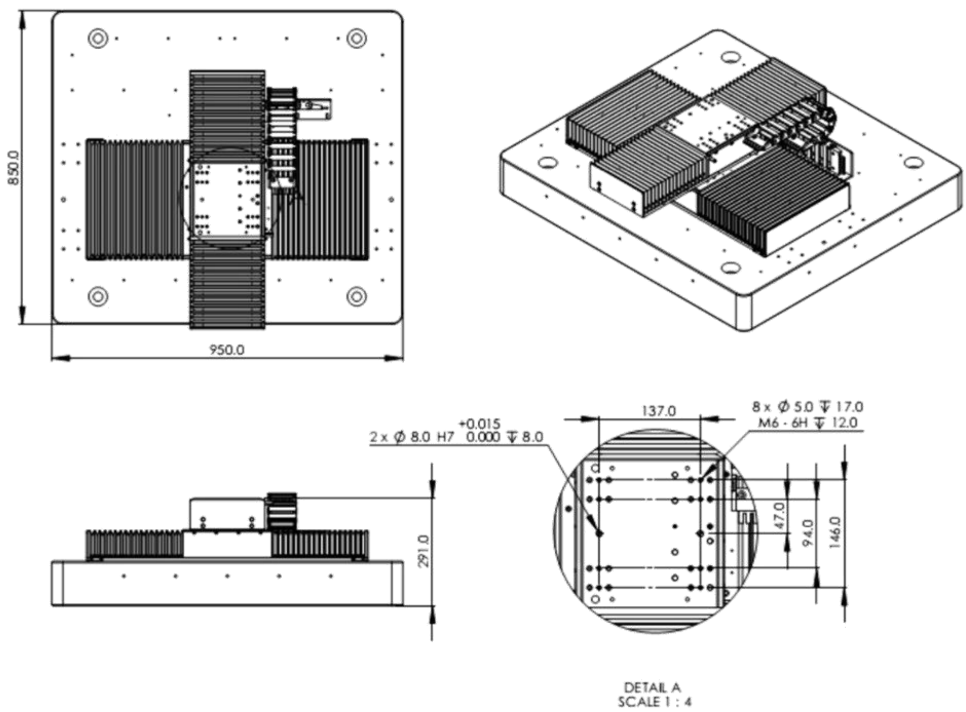


- Direct drive, no cogging, zero backlash linear motor
- Linear encoder options of 1µm, 0.5µm, and 0.1µm
- High accelerations (up to 10m/s^2) and speeds (up to 5m/s)
- Smooth motion at low speeds (low velocity ripple)
- Precise homing through encoder index pulse
- Full bellow cover

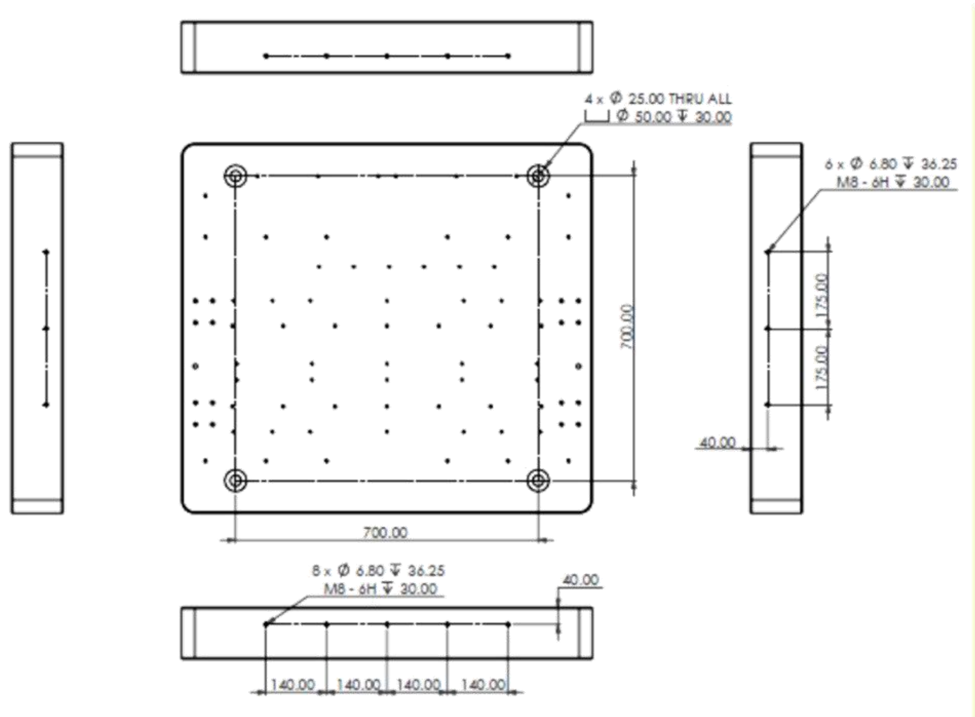
Motor Parameter

Specifications	Model	DGB300		DGBS300	
	Axis	Top	Bottom	Top	Bottom
	Motor	AUM4-S2	AUM4-S3	AUM4-S3	AUM4-S4
Performance Parameters	**Unit**	**Series**	**Series**	**Series**	**Series**
Continuous Force, coil @100°C	N	110	166	166	221.0
Peak Force	N	624	936	936	1248.0
Motor Constant	N/SqRt(W)	15.8	19.4	19.4	22.4
Continuous Power	W	48.7	73.0	73.0	97.3
Peak Power	W	1555	2332	2332	3110
Electrical Cycle	mm	60.0	60.0	60.0	60.0
Max Bus Voltage	V	330.0	330.0	330.0	330.0
Max Coil Temperature	°C	125.0	125.0	125.0	125.0
Thermal Dissipation Constant	W/°C	0.65	0.97	0.97	1.30
Continuous Current	Arms	2.3	2.3	2.3	2.3
Peak Current	Arms	13.0	13.0	13.0	13.0
Force Constant	N/Arms	48.0	72.0	72.0	96.0
Back EMF Constant	Vpeak/(m/s)	39.2	58.8	58.8	78.4
Inductance	mH	7.00	10.50	10.50	14.00
Terminal Resistance @25°C	Ohms	9.20	13.80	13.80	18.40
Electrical Time Constant	ms	0.76	0.76	0.76	0.76
Mechanical Parameters					
Coil Mass	Kg	0.56	0.89	0.89	1.19
Coil Length	mm	121.0	181.0	181.0	241.0

Dimension Drawing



Mounting Holes for Customer



Note: Additional tapped/through holes are possible upon request.

Effective Stroke

Model	Drive type	Axis	Effective Stroke (mm)	Moving Mass (Kg)	Total Mass (Kg)	Sensor Position (mm)	Hard Stopper Position (mm)
DGB300	N/A	Top	300	3.1	29	304	320
		Bottom	300	24.9	103	304	320
DGBS300	N/A	Top	300	3.4	29	304	320
		Bottom	300	26.1	103	304	320

Note that granite mass is approx. 260 Kg.

Performance Parameter

Specification Parameter	Unit	DGB300	DGBS300
[1]Straightness	µm	±3µm/25mm NTE±10µm/300mm	±3µm/25mm, NTE±10µm/300mm
[1]Flatness		±3µm/25mm NTE±10µm/300mm	±3µm/25mm, NTE±10µm/300mm
Repeatability (1µm resolution)		±3µm (40µm scale pitch)	±3µm (40µm scale pitch)
Repeatability (0.5µm resolution)		±1.5µm (20µm scale pitch)	±1.5µm (20µm scale pitch)
Repeatability (0.1µm resolution)		±1µm(20µm scale pitch)	±1µm(20µm scale pitch)
Repeatability (Analogue)		±5 counts	±5 counts
X-Y Orthogonality	Arc-sec	10	10

Part Numbering (Rapid Delivery Series)

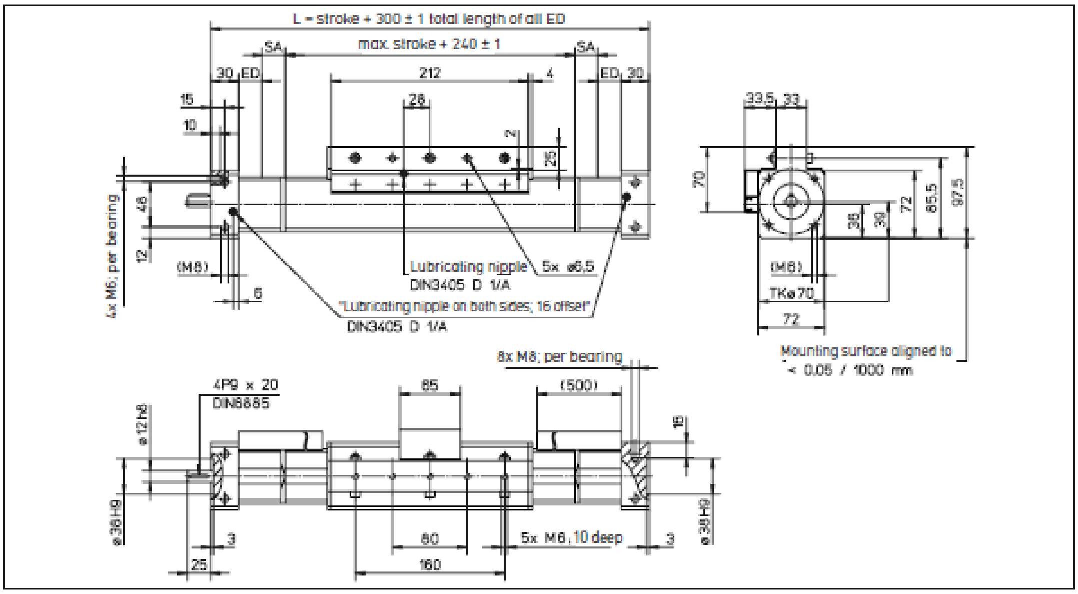
	Stroke	Top Axis Motor Model	Bottom Axis Motor Model	Sensor Type	Hall Option	[2]Motor Cable (m)	Encoder Option (type)	Encoder Resolution (micron, µ)	Rail
DGB DGBS	300	AUM4-S-S2 AUM4-S-S3	AUM4-S-S3 AUM4-S-S4	J K	H9D	1.0-5.0	R22, [3]R10	0.1 0.5 1	T

1. All measurement taken when module is mounted on a 5 micron flat granite table.
2. Motor cable is the length of cable measured after the bottom axis carriage
3. Available for 1 um only.

Example: DGB300-AUM4-S-S2-AUM4-S-S3-J-H9D-3.0-R22-0.1-T

Linear ball screw actuator technical parameters

WIESEL® W02
with ball screw drive and sliding guide

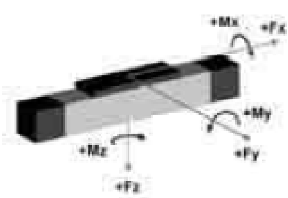


Technical data

- Linear speed: max. 1 m/s
- Repeatability: ± 0.05 mm
- Acceleration:
  Single lead nut max. 5 m/s²
  Double lead nut max. 10 m/s²
- Rotational speed: max. 3000 rpm
- Drive element: Ball screw drive
  Diameter: 20 mm
  Lead: 5 or 20 mm
- Stroke length: 40 up to 5200 mm
- Power bridge: 212 mm long
- Geometrical moment of inertia: Iy $6.52 \cdot 10^5$ mm⁴, Iz $5.99 \cdot 10^5$ mm⁴
- Weights
  Base without stroke: 3.60 kg
  100 mm stroke: 0.70 kg
  Carriage: 0.60 kg

Trapezoidal screw drive on request (4, 8 and 16 mm pitch)

Load and load moments



Load	dynam. [N]
Fx drive 2005	2500
Fx drive 2020	1500
Fy	500
±Fz	650

Load moment	dynam. [Nm]
Mx	30
My	70
Mz	50

Order code see page 126

Idle torques [Nm]

Rotational speed [rpm]	Lead P [mm]			
	MM 5	MM 20	M 5	M 20
150	0.75	1.00	0.50	0.70
1500	1.30	1.50	1.00	1.35
3000	1.75	2.00	1.50	1.80

Additional length with spindle support and end dampers

Length [mm]	L_{SA} [mm]	L_{ED} [mm]
30	4 SA	–
60	6 SA	–
70	–	2 ED

SA-diagram