

ELECTRIC VEHICLE BATTERY MODULE DISMANTLING

Analysis and Evaluation of Robotic Dismantling Techniques for Irreversible Fasteners, including Object Detection of Components.

Orwah Zubaidi
Jostein Tjerbo Korneliussen

SUPERVISOR
Martin Marie Hubert Choux
Muhammad Talha Bilal

University of Agder, 2022
Faculty of Engineering and Science
Department of Engineering and Sciences

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja
2.	Vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none">• Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.• Ikke refererer til andres arbeid uten at det er oppgitt.• Ikke refererer til eget tidligere arbeid uten at det er oppgitt.• Har alle referansene oppgitt i litteraturlisten.• Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	Ja
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høyskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja
7.	Vi har i flertall blitt enige om at innsatsen innad i gruppen er merkbart forskjellig og ønsker dermed å vurderes individuelt. Ordinært vurderes alle deltakere i prosjektet samlet.	Nei

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).

Opgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja
Er oppgaven båndlagt (konfidensiell)?	Nei
Er oppgaven unntatt offentlighet?	Nei

Acknowledgements

We would like to express our special thanks Associate Professor Martin Marie Hubert Choux and PhD research fellow Muhammad Talha Bilal for providing their invaluable guidance and assistance throughout the duration of the project. Their contributions are sincerely appreciated and gratefully acknowledged.

Further, We would like to thank the senior engineers Cecilie Ødegård, Harald Sauvik and Roy Werner Folgerø for being helpful with lab experiments. We would also like to thank the University for the disposition of equipment from the lab, as well as their expertise within different disciplines.

An appreciation is also reserved to our friends and family for their love and support.

Special thanks to:

Martin Marie Hubert Choux , Associate Professor, University of Agder
Muhammad Talha Bilal, PhD Research Fellow, University of Agder

Without you this project would have been a lot more difficult to complete.

Orwah Zubaidi

Jostein Tjerbo Korneliussen

Abstract

This thesis examines a study of The Lithium-Ion Battery (LIB) from a electric vehicle, and it's recycling processes. A Battery Module (BM) from the LIB is shredded when considered an End-Of-Life product, and motivates for automated dismantling concepts to separate the components to save raw materials.

From State-of-the-art (SoA) research projects and background theory, automatic module dismantling concepts have been evaluated for a Volkswagen E-Golf 2019 battery module. The presence of irreversible fasteners make the use off destructive dismantling techniques necessary. This study evaluates two different concepts to disconnect laser welds holding together the compressive plates made of steel. A hydraulic actuated concept is first investigated to separate the welded compressive plates within the casing. A FEM analysis with different configurations is performed to evaluate the most effective hydraulic solution when analysing the Von Mises stress. This solution is further compared with another automatic dismantling concept, namely milling. For the purpose of an automated milling concept, manipulators from ABB are assessed and the feasibility is verified based on results from manual milling operation.

The proposed dismantling operation is made possible by developing a system architecture combining robotic control and computer vision. Open source software based on Robot Operating System (ROS) and MoveIt connect and control an ABB IRB4400 industrial robot whereas the computer vision setup involves a cutting edge 3D camera, Zivid, and object detection algorithm YOLOv5 best suited for this task. Adjustable acquisition settings in services from Zivid's ROS driver are tested to accomplish the optimal capture configuration. Two datasets generated with RoboFlow were exported in the YOLOv5 PyTorch format. Custom object detection models with annotated components from the BM was trained and tested with image captures.

All in all, this study demonstrates that the automatic dismantling of battery modules can be achieved even though they include irreversible fasteners. The proposed methods are verified on a specific battery module (Egolf 2019) but are flexible enough to be easily extended to a large variety of EV battery modules.

GitHub: <https://github.com/JoKorn97/MAS500-Group1>

Contents

Acknowledgements	ii
Abstract	iii
1 Introduction	1
1.1 Background	1
1.1.1 The Litium-Ion battery in EV's	1
1.1.2 Recycling Lithium-Ion Batteries	2
1.2 Project objective	4
1.2.1 Research questions	4
2 State of the Art & Background theory	5
2.1 State of the art	5
2.1.1 Task planner	5
2.1.2 Module disassembly	6
2.2 Industrial-robot control	7
2.3 ROS	9
2.4 Finite Element Method	11
2.5 Camera technology	13
2.6 Object detection	14
3 Module Disassembly Concepts	17
3.1 Volkswagen E-Golf 2019, battery module	17
3.2 Connection diagram	19
3.3 Laser Beam Welding	19
3.3.1 Tension test	21
3.4 Dismantling Concepts	23
3.4.1 Fracture of the laser weld	23
3.4.2 Milling	26
4 Experimental setup	30
4.1 ABB IRB 4400/60 + ABB IRBT 4004	30
4.2 Zivid One+ Medium	31
4.3 Computer hardware	31
4.4 System architecture	32
4.5 UiA Robotics lab setup	32
5 Image acquisition	38
5.1 Zivid ROS driver	38
5.2 Acquisition settings	40
5.3 Image capture	44

6	Object detection	50
6.1	YOLO	50
6.1.1	Annotation & dataset generation	52
6.1.2	Training	53
7	Project Results	60
8	Discussions	68
9	Conclusion & Further work	70
A	ABB IRB 4400 datasheet	73
B	ABB IRBT 4004 datasheet	87
C	Zivid One+ datasheet	90
D	Move Group python interface test	112
E	Software workflow	114
	Bibliography	119

Chapter 1

Introduction

In the recent years there has been a major change in the car fleet all around the world. The continuous concern about climate change and pollution has forced the biggest car manufacturers to see the value in electric cars. In 2021, 64.5% of new registered cars in Norway have been electric cars, and its estimated that we may already have half a million electric cars in the norwegian car fleet by March 2022 [50].

1.1 Background

This thesis is a part of the research project Lithium Ion Battery Recycling (LIBRES). The LIBRES project is owned by Hydro. It's a ongoing project which started in 2018, and planned to be ended by 2022. LIBRES is financed by the Norwegian Research Council. There are several companies and universities included in this project, such as: BatteriRetur, Glencore Nikkelverk, Elkem Carbon, and University of (UiA) [18].

1.1.1 The Litium-Ion battery in EV's

Different approaches of electrifying cars have been proposed by car manufacturers. A wide variety of different configurations, cell types and cell chemistries exists in the market due to the car manufacturer's different approaches. Three main components which are used between the manufacturers are: battery pack, -module and -cell. However, their physical design and configuration differ. Their differentiation in physical design empowers challenges within automated battery recycling [6]. As an example, three different battery packs from different manufacturers (2014) are introduced in Figure 1.1.

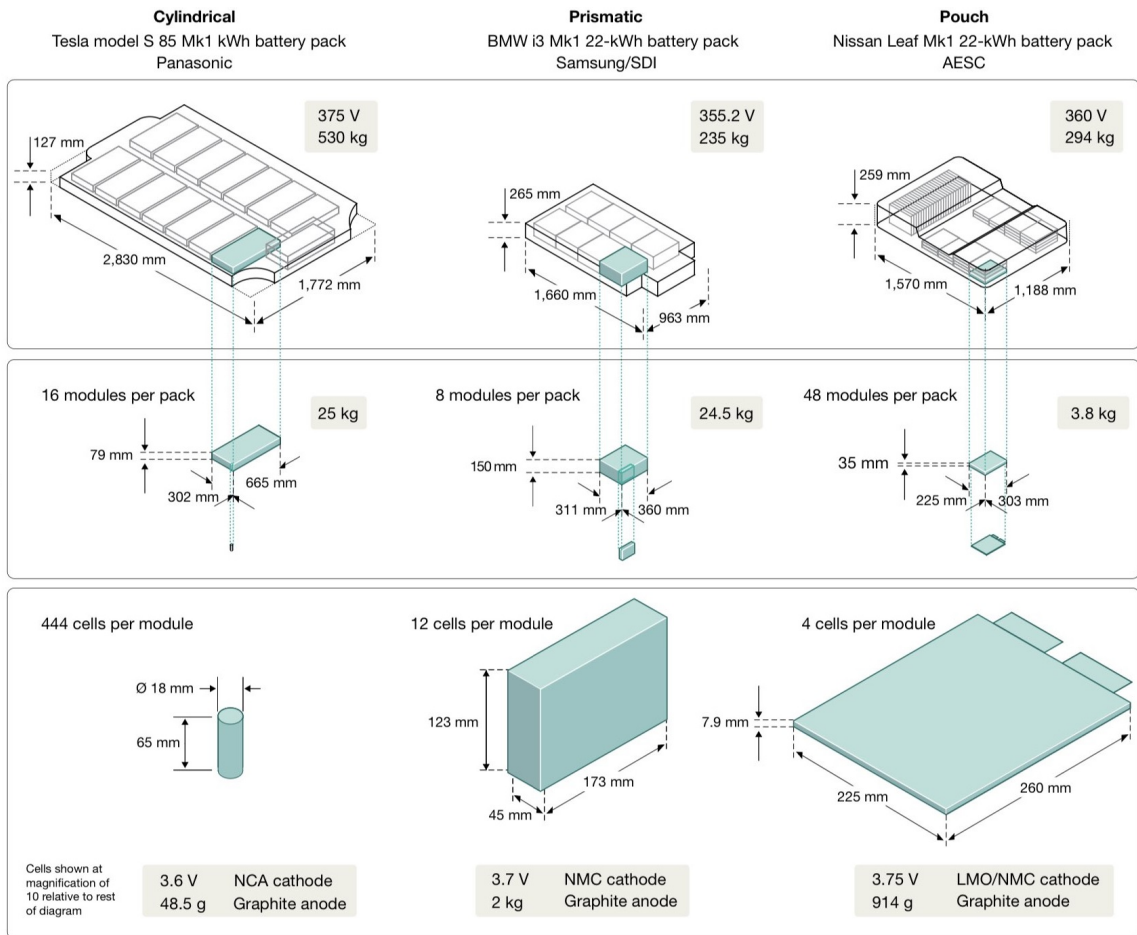


Figure 1.1: Breakdown of different battery packs, from different manufacturers [6]

1.1.2 Recycling Lithium-Ion Batteries

To characterize the functionality of a Lithium-Ion battery (LIB) an assessment of the "State of health" degree has to be considered [6]. If the battery does not meet within its initial design specifications, the battery motivates for either dumping or recycling and the battery is characterized as a end of life (EOL) product. One considerable bottleneck in the recycling process of LIB is the manual disassembly which leads to high costs and safety risks. Due to limited standardisation and large variety of models its challenging to automate the process. Automated solution would need the most recent advancements in vision, robotics, instrumentation and AI-based planning and acting. Some knowledge gaps are still to be filled, which motivates for a automated disassembly process in the future. An automated disassembly process is considered mandatory due to the predicted quantities of returned battery systems [9].

There are mainly four different recycling processes utilized for LIBs [29]:

- Pyrometallurgical
- Hydrometallurgical
- Bioleaching
- Direct recycling

Pyrometallurgical recycling process of LIB's involves thermal treatment of minerals, ores, and concentrates, which allows recovery of valuable metals. The pyrometallurgical recycling method produces high volumes of steels and iron, and is mainly employed in industrial-scale applications due to its simplicity and high productivity. The pyrometallurgical process is mainly divided into two approaches; (1) Through a heat-treatment process which regenerates electrode materials by lithiation or crystals repairs, (2) Acquire Fe-, Ni-, Co- and Mn-based liquid alloys at a temperature higher than 1000°C from LIB [57]. Several companies have adapted the pyrometallurgical recycling process.

The Hydrometallurgical method in short, uses a chemical solution where the water is the solvent to recover the metals. This method has showed a success rate of 95% for recovering cobalt (Co) and lithium (Li) from the Lithium-Ion battery and is described in detail in [79].

Bioleaching refers to the use of microorganism to extract metals from their ores. Bioleaching is famously used by the mining industry, due to its efficiency and environmental friendliness. This technique is cost- and energy-effective and produces less dangerous by-products [37].

Ideal recycling process: In the current recycling process today, the modules are shredded and needs an energy consuming chemical process to produce useful materials. This method is waste full, inefficient and increases the carbon print on the environment. This goes against its purpose if electric cars are to be environmentally friendly [6]. Figure 1.2 compare an ideal-versus the real (direct) recycling process.

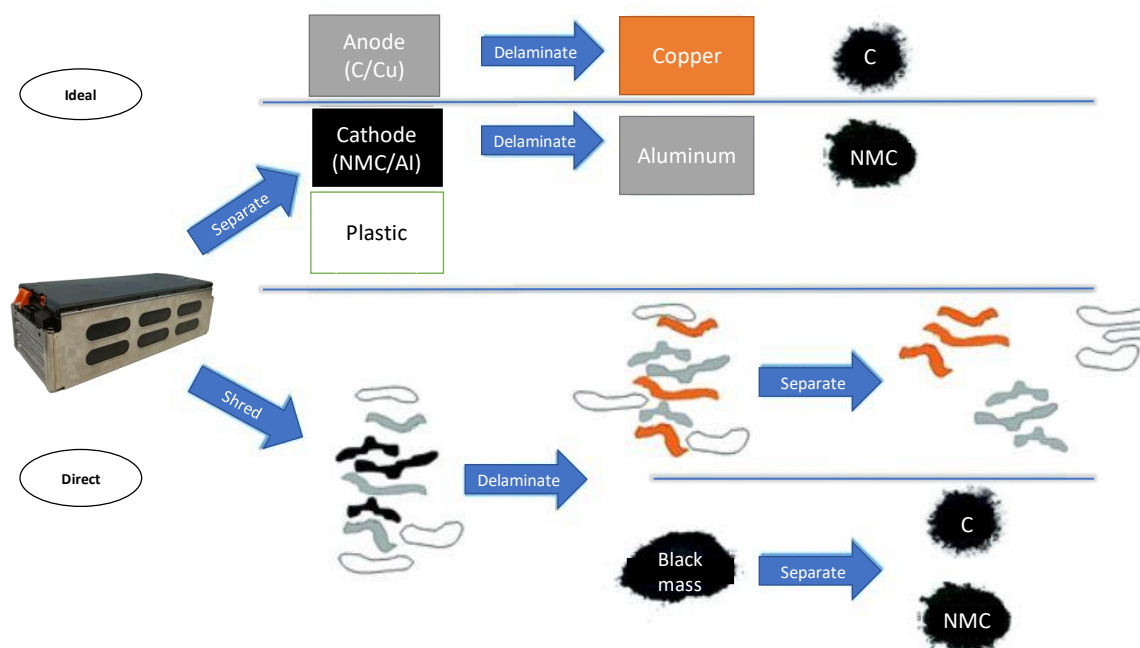


Figure 1.2: Ideal vs direct battery recycling process [5]

Today, the disassembly process of Lithium Ion batteries are done manually. The process requires two high-voltage certified technicians, and takes approximately 45 minutes. UiA's ambition for the project is to automate parts of the manual disassembly process, and improve its efficiency. The main goal of the LIBRES project is to develop technologies for an Electric Vehicle (EV) battery recycling plant in Norway. Then to be utilized in a pilot plant planned for start-up in 2024 [18].

Another project worth mentioning is the ReLieVe project (Recycling Li-ion batteries for Electric Vehicle). This is a European collaboration project between different companies and universities including Eramet, Suez Group, Chimie ParisTech and Norwegian University of Science and Technology (NTNU) [25]. Additionally, Northvolt (Swedish battery supplier) and Hydro (Norwegian energy and aluminium company) established Hydrovolt to deliver sustainable battery recycling in Europe. Hydrovolt recycles copper, plastics, aluminum, and a substance known as black matter, which contains important metals such as nickel, manganese, cobalt, and lithium. [30]. The main goal of these project is to develop a recycling process for Li-ion batteries used in electric vehicles and recover the elements to be used in production of new batteries.

1.2 Project objective

UiA has provided research material for LIBRES through other bachelor- and master thesis'. The main goal of this project is to integrate the previous contributions to analyse and evaluate the VW E-Golf mark 2 (mk2) battery module (BM). The research incorporates component analysis of the BM, dismantling evaluation, computer vision and artificial intelligence (AI). Robot Operating System (ROS) is utilized to merge image acquisitions and robot control using python. The project is divided to the following subgoals:

1. Study the disassembly process of the BM
2. Research potential robotic dismantling techniques for the disassembly process.
3. Develop a configuration for robotic control from a personal laptop.
4. Train computer vision algorithm and evaluate camera parameters to recognize components and/or fasteners.

1.2.1 Research questions

- Is it possible to disassemble the battery module's compressive plates without using cutting tools, and if so, what is the best alternative?
- Is it possible to automate the disassembly process for the purpose of recycling ?

Chapter 2

State of the Art & Background theory

This chapter is intended to provide necessary background information, and research other relevant projects in order to solve the project objective. The theory will also be used to defend choices/solutions through the project.

2.1 State of the art

Due to the forecasted quantity of LIB in EV, an automated disassembly plant is considered mandatory for future recycling of EVB's. This section will present some previous work of automated EVB recycling.

2.1.1 Task planner

Choux et. al. [17] introduced a task planner for robotic disassembly of electric vehicle battery packs. The experimental setup included a computer with Ubuntu and the Robot Operating System (ROS) connected to a manipulator from ABB, a linear track from ABB, and a RGBD camera from Zivid. A Lithium-Ion battery pack from an Audi sportback e-tron was utilized for the experiment as well. Their proposed concept of the task planner can be seen in Figure 2.1.

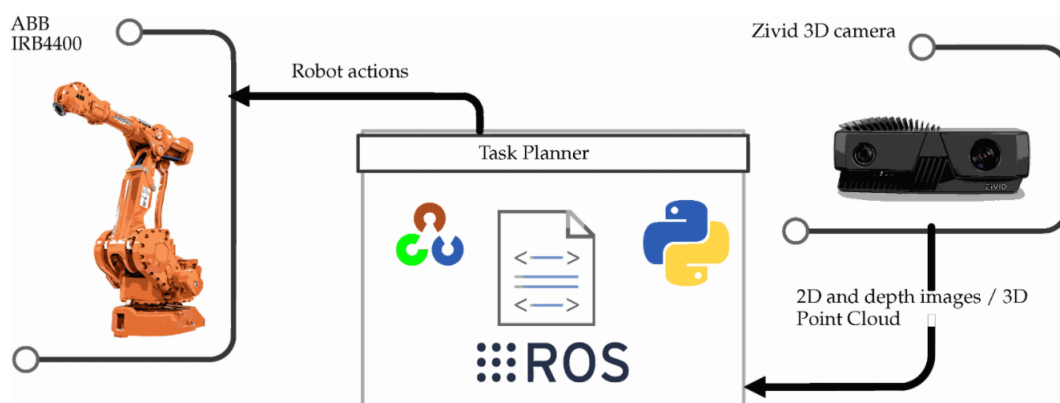


Figure 2.1: Proposed concept from Choux et. al for their task planner, [17]

With information provided from the data given by the Zivid camera, the task planner could be able to make decisions regarding disassembly of the battery pack, and control the manipulator respectively. The main loop of their proposed concept presented in steps:

- **Step 1** Take 2D and 3D images
- **Step 2** Detect and identify components

- **Step 3** Within the world frame, find the position of the detected component.
- **Step 4** Order of operation
- **Step 5** Remove component
- **Step 6** Until state goal is reached; repeat

As an end result, the accuracy of the system was tested. The experimental test was performed with a 3D-printed pointer used as a Tool Center Point (TCP). From several tests, a mean accuracy of <5mm from the desired position (screw/bolt) was acquired.

2.1.2 Module disassembly

Tan et al. [70] proposed a disassembly process for electric vehicle batteries (EVB's) in 5 steps:

- **Step 1** Test the battery if for residual charge. If there is, discharge is necessary.
- **Step 2** Remove screws and adhesive to dismantle the battery casing.
- **Step 3** External components of the battery pack has to be removed. Components such as: Battery management system (BMS), power electronics and heating-/cooling system.
- **Step 4** Cut or removed additional wires, cables and connectors.
- **Step 5** Disassemble the stack holders, which provides support for the battery modules in the battery pack. Remove the stack holders, and the BM's can be extracted. Which then can be disassembled to battery cells.

Nguyen [44] researched a complete automated dismantling line for BM's. The project included a review, verification and evaluation of a designed BM dismantling line.

The research provided a thorough analysis for some BM's from different EV manufacturers. A discharged BM was tore down part by part, and he assigned the material and quantity of each part. During the manual disassembly analysis, Nguyen also documented what specific tool which was required for each dismantling operation. Fastening types vary between the design of the BM's, and Nguyen provided data of different types of fasteners which requires different types of dismantling techniques.

From the data provided by the BM analysis, Nguyen [44] proposed automated concepts which could be utilized as disassembly workstation for BMs. Each concepts were evaluated with pros and cons, and a disassembly requirement table. The disassembly concepts were weighted from 1 to 5 by these five criteria:

- Dismantling time
- Cost of manufacturing
- Cooperation with other stations
- Potential for update/expand
- Flexibility and robustness

Concept No.	Workstation	Robot cell	Vision System
1	Rotary worktable and fixture system for BM	1 manipulator, required tools for disassembly and a tool bank	1 stationary camera
2	Stationary worktable and fixture system for BM	2 manipulators, required tools for disassembly and a tool bank	1 stationary camera

Table 2.1: Proposed concepts from [44]

From the concept evaluation, concept No.1 and concept No.3 of five proposed concepts were utilized for further simulation. Concept 1 and 2 is listed in Table 2.1

The two concepts were then simulated utilizing RoboDK. RoboDK is a software for offline programming and simulation of manipulators and controllers [58]. Nguyen developed the concepts in RoboDK, and performed a comparison between the two chosen concepts. A BM from VW E-Golf 2013 was utilized. From the experiment, the time lost in tool change was significant. Concept 1 lost 70 seconds from tool change operations, while this lost time was avoided with Concept 3 due to it's capacity with two manipulators.

2.2 Industrial-robot control

In general, the research in robotics is advanced and there is more than 500 companies in the world working in the robotic sector [15]. The robotic sector can be divided into six main categories: industrial robotic arms and collaborative robots (cobots), humanoid robots, autonomous mobile robots (AMRs), automated guided vehicles (AGVs), and hybrids.

In this project, the main focus is the conventional industrial robots since the IRB4400 is the robotic arm to be used for the tasks in hand. Usually robotic companies develop their own software and interface for robotic control. ABBs well known RobotStudio is a great tool for simulation, path planning and offline programming, but the lack of compatibility with sensors from other vendors and companies increases the complexity. The interface in this project is developed using Robot Operating System (ROS), described in 2.3.

Robot Kinematics and Dynamics

In order to precisely control a robotic system, the geometry of the robots is utilized to study the movement of kinematic chains. Signifying that each link of the robot are represented as a rigid body and the joints are defined by the properties they have. Kinematics is defined as a the study of the motion (position, velocity and acceleration) of objects without taking into account the cause of motion e.g. the forces and torques.

Dynamics is the study of the relationship between change in motion and the cause of that change. Newton's laws of motion are deeply involved in dynamics as it is a part of classical mechanics. This study allows us to compute the forces and torques needed for the desired motion, were mass and inertia properties are defined.

The inverse- and forward dynamics problem is usually solved and integrated in the robot controller. The actuation forces and torques for any desired motion will not be studied further in this thesis.

Transformation Matrix

The position of the end-effector is the sum of all vectors representing the links in the kinematic chain. Its essential to express all vectors in the same coordinate frame when performing the necessary calculations. This is done by using a transformation matrix to represent any coordinate frame within a global coordinate system, specifying the position and orientation of the local coordinate system [12].

Homogeneous transformation matrix 2.1 have the advantage of combining translation and rotation into a single matrix. This makes 3D transformations easier to perform. Aside from splitting rotation and translation into two components. The matrix is constructed in four dimensions. The matrix is made up of a rotation component 3x3 in the upper left corner, a translation component 1x3 in the rightmost column, and a constant whose value does not change. Fig. 2.2 illustrates how a local frame is translated to point P relative to a global frame.

$$H = \begin{bmatrix} n_x & o_x & a_x & Px \\ n_y & o_y & a_y & Py \\ n_z & o_z & a_z & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

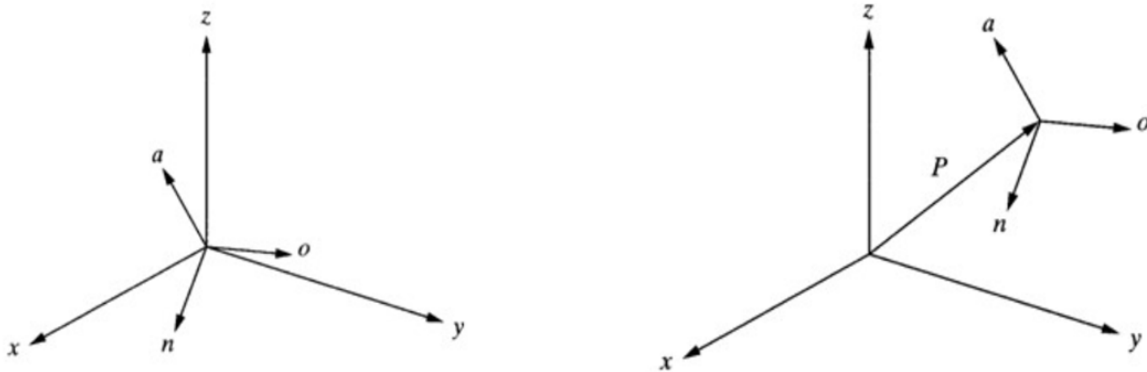


Figure 2.2

Forward- and Inverse Kinematics

The goal in forward kinematics is to calculate the position of the end effector using kinematic equations, were the joint space variables (angles of each joint and the length of each link) of the robot is known. Inverse kinematics is the opposite of forward kinematics, the goal is to calculate the joint space variables using the cartesian space variables (the position and orientation of the end effector), Fig. 2.3 illustrates this process. When inverse kinematics is utilized, commanding the robot to move to a specific point in space becomes less challenging, but not completely without limitations.

Theoretically the robot can reach a given point in its workspace with infinite number of configurations. It is called redundancy, and it is one of the challenges in robotic control. Normally, this occurs when the robot arm has more DoF than it needs to reach a given point. This is solved with computer algorithms such as KDL (kinematics and dynamics library), explained in chapter 2.3.

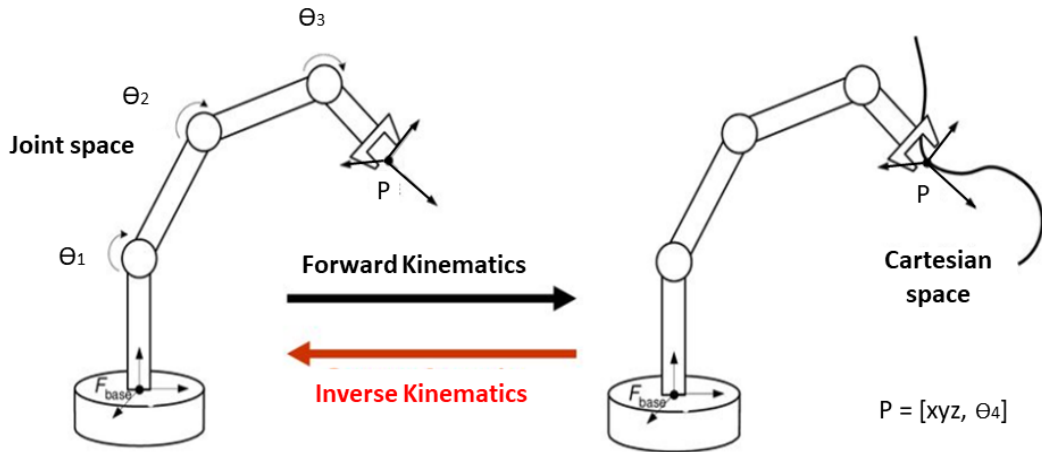


Figure 2.3: Forward- and Inverse Kinematics

2.3 ROS

ROS [56] (Robot Operating System) has received increasing attention in the robotics community in recent years. ROS is a collection of open source software framework that serves as a middleware between robots, developed applications and the actual operating system [8]. ROS allows any robot, regardless of brand, to work collaboratively. Many programs for common robotic functionalities and algorithms have been developed by the open source robotics research community based on the ROS-framework. ROS provides essential services for communication, message exchange, software package management and widely used functionalities. ROS is heavily reliant on essential libraries for robot control, simulation and path planning.

Fig. 2.4 illustrates the overall structure of ROS. The system communicates by publish/-subscribe pattern, the master coordinate the messages between different nodes and handles registrations of nodes. Sensor information or joint states are examples of topics that nodes can publish- or subscribe to if the information is needed. The information is wrapped in messages defined according to the convenient data type.

- **Master:** Responsible for registration and synchronization between nodes, topics and services.
- **Topic:** Message delivering system, used for continuous data streams (sensor information).
- **Service:** Message delivering system, used for remote procedure calls that have a short duration (inverse kinematic calculation).
- **Message:** Data that is being broadcast on a specific topic.
- **Node:** Executable program within an application. Publish/subscribe to a topic.

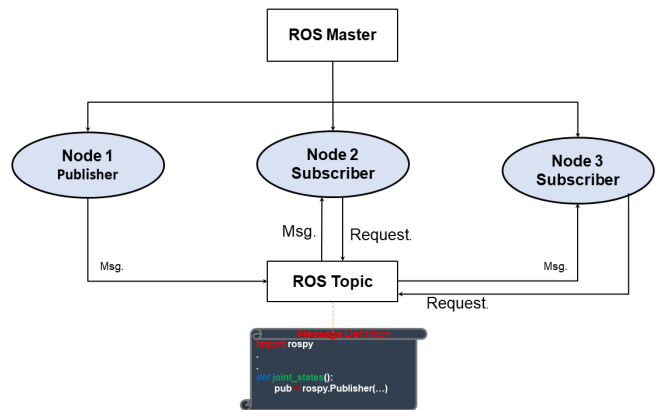


Figure 2.4: Overview of ROS structure

MoveIt

MoveIt is an open-source application implemented with ROS for simulation, manipulation and control of a robot. MoveIt provides a framework for developing the applications functionality by establishing a code base for implementing essential features of the robot application. MoveIt utilizes the structure of ROS and contains various of libraries including OMPL (open motion planning library) and KDL (kinematics and dynamics library). MoveIt can be integrated with ROS Visualization (RViz) in order to visualize path planning and simulation of a desired path with a robot [19].

The primary node `move_group` in MoveIt's architecture acts as an integrator, bringing all of the separate components together to provide users with a set of ROS actions and services [39]. From Fig. 2.5, `move_group`'s connection diagram can be seen.

As shown in Fig. 2.5, the input *ROS Param Server* is provided from a specified MoveIt configuration and can be generated from the graphical user interface (gui) *MoveIt Setup Assistant*. The MoveIt Setup Assistant is a gui that allows you to configure any robot for MoveIt. Generating a Semantic Robot Description Format [64] (SRDF) is its main purpose. While configuring a robot in the Moveit Setup Assistant, there is several steps which provides necessary data to generate the configuration [61]. Three of the steps are listed below:

- **Generation of self collision matrix**, seek for pairs of connections on the robot that can be securely disabled from collision checking.
- **Add virtual joints**, to connect with the world frame.
- **Add robot poses**, configure fixed poses.

`moveit_commander` (Python) is one of the user interfaces which can execute path services to the robot controller, and are considered on of the most user friendly [60].

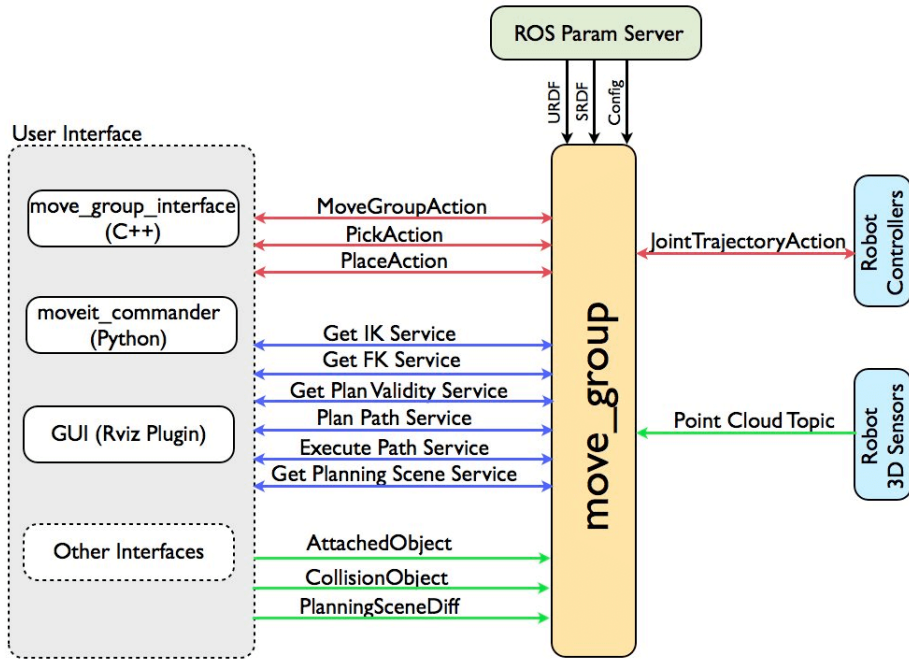


Figure 2.5: Connection diagram `move_group` [39]

2.4 Finite Element Method

The finite element method is used in numerical analysis to solve partial differential equations numerically. These can, for example, be used to represent the dynamic behavior of certain physical systems analytically [26].

The finite element method (FEM) divides complex structures into simple elements (parts) with known and understood properties. The system can then be rebuilt using these elements to study the overall reactions to the properties of the building elements and how they interact. The method is used primarily in engineering science. A variety of CAD-software that have simulation tools for FEM-analysis, such as Solidworks, Abaqus and Ansys are utilized. Analysing the designed model begins with generating a CAD-model. The model is then divided into elements connected by nodes and represented as a network of discrete interconnected elements.

Linear static analysis

The first condition of static analysis is that force is static and does not vary with time, the second is that the sum of moments and forces equals zero. This condition must be met by the simulated model at every node. The total sum of the external forces and moments in the model is equal to the reaction forces and moments. The equation 2.2 to be solved is as follows:

$$KD = F \quad (2.2)$$

where:

K is the stiffness matrix

D is the nodal displacement vector

F is the external torques and forces vector

Mesh: Meshing is the process of dividing the structure into smaller elements. The mesh is depending on the element type which can be 1D, 2D or 3D as shown in Fig. 2.7. Higher mesh element concentration is typically used near sharp corners or smaller geometrical regions. The accuracy and computational time increase as the mesh density increases. Sharp corners in a model will produce a stress singularity, resulting in an inaccurate prediction.

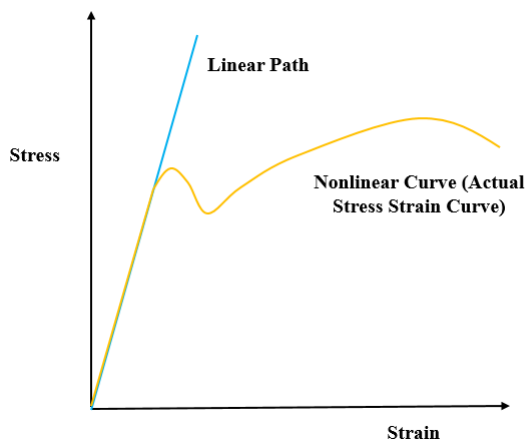


Figure 2.6: Linear static analysis [24]

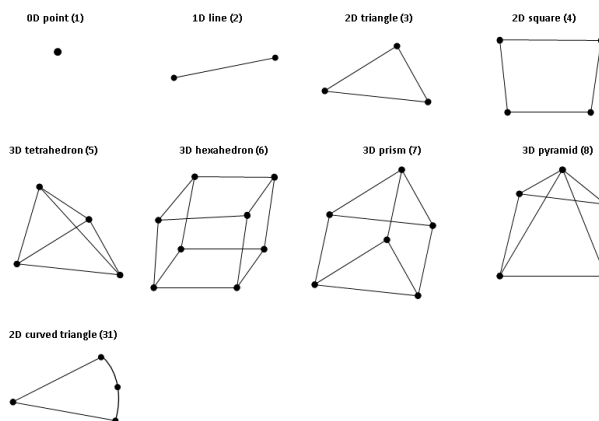


Figure 2.7: Element Types

2.5 Camera technology

A device that captures light information from its surroundings, converts it into a usable physical quantity, and visualizes the data map as seen from its perspective is called a camera. In short term: a camera converts 3D objects in world into a 2D image, i.e. perspective transformation. It's considered a *non-contact* and *optical* device, since it does not interact physically with the surroundings but takes advantages of the light conditions. Fig. 2.8 is a simplified representation of the concept *pin-hole-model*.

A pinhole allows light from the surroundings to enter a lightproof chamber. The small size of the hole, ideally a point, prevents light from dif-fusing. Projected light which enters the chamber hits the opposite side in a straight line and make a projection of the environment [28].

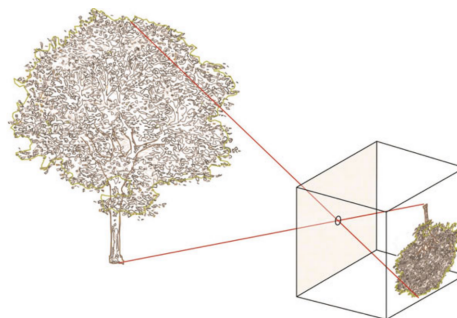


Figure 2.8: Simplified pin-hole-model [28]

However, a 3D camera applies the concept of a camera includes depth measurement providing the third dimension. Depth cameras have two key drawbacks: their field of view is limited, and they can only retrieve a depth field rather than genuine 3D data [20].

Triangulation and Direct depth estimation are two 3D camera measurement principles. Their main difference is that triangulation methods utilize two cameras, while direct depth estimate make use of 1 [28]. As seen in Fig. 2.9, the orange boxes are the 3D shape of interest, as seen by the camera with its field of view displayed in blue.

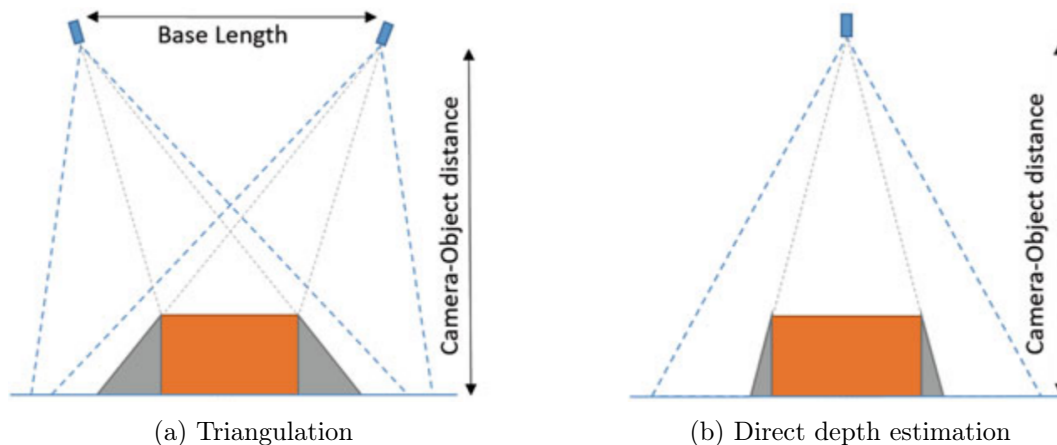


Figure 2.9: 3D measurement techniques [28]

Two common types of depth camera technologies are *Time-of-flight* and *Structured light* [80].

Structured light depth camera

Structured-Light cameras employ a single camera to display a structured pattern into the scene. Instead of using two cameras to triangulate, a laser projector is used instead. It projects a defined pattern with enough structure to create unique correspondences for the camera to triangulate. The camera, which can triangulate based on the structured pattern, knows the direction of the pattern [28]. The structured pattern generated from each light value can be modeled as a active pin-hole system. Instead of receiving light as in standard cameras, light beams linking the center of projection and the scene point P through pixel pa are emitted [80].

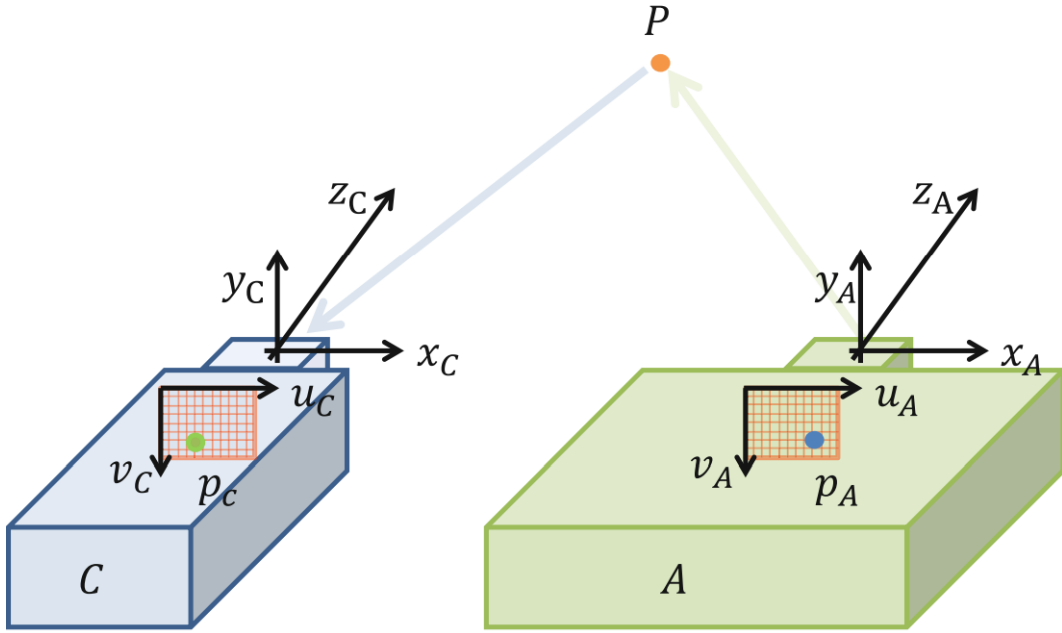


Figure 2.10: Active triangulation, with camera C and illuminator A [80]

Laser triangulation is the most basic structured-light system, Fig. 2.11 illustrates a single laser triangulation. On a scene or an object, a laser beam is projected. The camera locates and recovers the position and depth of the dot on the scene from Equation 2.3 and 2.4.

$$z = \frac{b}{\tan(\alpha) + \tan(\beta)} \quad (2.3)$$

$$x = z * \tan(\alpha) \quad (2.4)$$

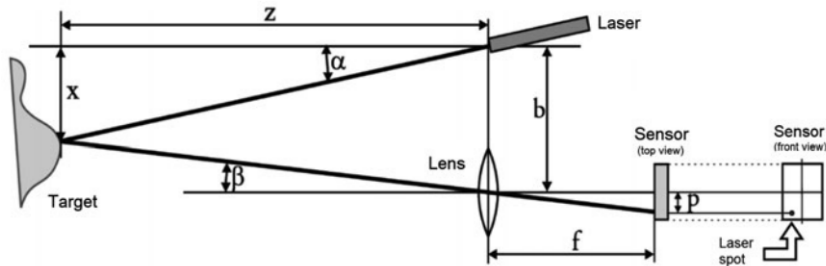


Figure 2.11: Single laser triangulation [28]

2.6 Object detection

One should not only concentrate on classifying distinct images to acquire a thorough comprehension of them, but also attempt to properly estimate the concepts and locations of objects perceived in each image [82]. Vehicle detection, face detection, text detection, and other applications of object detection are fundamental in scientific study and industrial production [7]. From Fig. 2.12, there is primarily 3 types of detection techniques to estimate estimate objects from an image:

- **Classification:** Classifies images into a single category of interest. Can include localization.

- **Object detection:** Detects multiple defined objects from an image. The desired object to be detected gets marked with a bounding box.
- **Instance segmentation:** Similarly to object detection. However, instead of an bounding box it recognizes the vertices around the desired object to be detected.

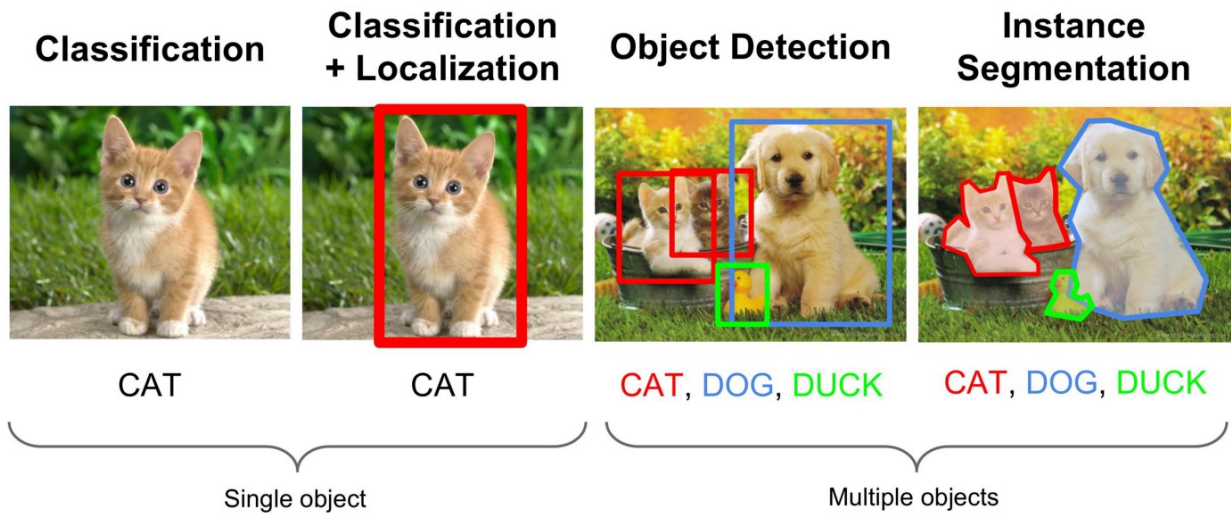


Figure 2.12: Detection methods [53]

A task like Object detection is applied in computer vision, and make use of Convolutional Neural Networks (CNN). CNN are made up from neurons. Each neuron has a specific weight, and bias. Connecting each neuron, the CNN-architecture contains several layers. Primarily an input layer, an output layer, and hidden layers. A convolutional layer, a pooling layer, a fully connected layer (FC), and other normalization layers make up the hidden layer. A convolution method is performed by the convolutional layer. Convolution in object detection is a method used to generate an output function from a filter and data/image input. The pooling layer reduce the dimension of the input. Every neuron in one layer is connected to every neuron in another layer by the FC layer [22]. A visualization representation of a convolutional- and fully connected layer can be seen in Fig. 2.13.

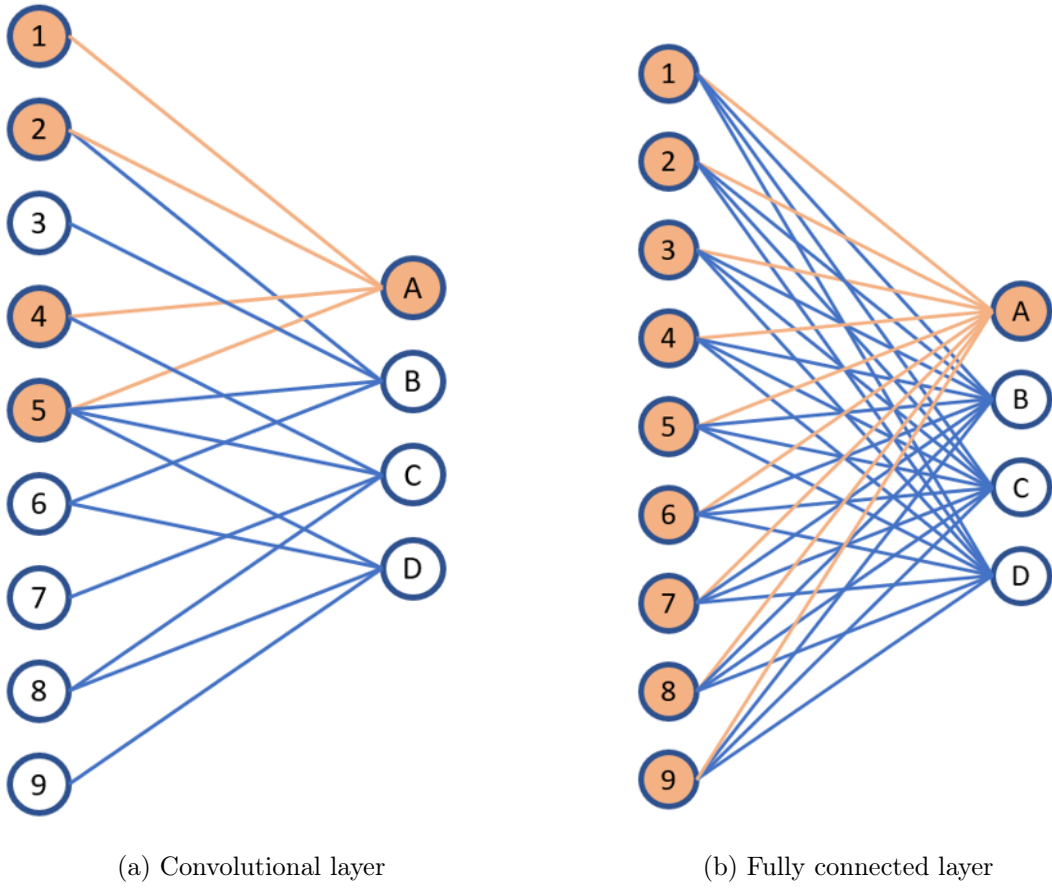


Figure 2.13: Convolutional - vs. fully connected layer [76]

CNN in object detection is mainly divided into two categories, *region proposal-based methods* and *classification-based methods*. Region based object detectors include R-CNN and Faster R-CNN. While classification based methods in You Only Look Once (YOLO) and single shot detector (SSD) [22].

Chapter 3

Module Disassembly Concepts

This project shifts focus away from the complete Electric Vehicle (EV) battery pack and narrows it down to the battery modules (BM). BatteriRetur AS provided 10 battery modules from a Volkswagen (VW) E-Golf which will be used in this project. This chapter presents further tear-down of the BM from VW.

3.1 Volkswagen E-Golf 2019, battery module

The BM from VW used in this project is manufactured by Samsung SDI. Information about these battery modules are not extensive, however some re-sellers provide necessary data [10]. Each BM from the 2019 E-Golf shown in Figure 3.1 contains 12 cells with a nominal voltage of 3V and electric charge of 3.8V. Three cells are connected in parallel, which provides 4 module cell packs connected in series. Electrical specifications of the BM can be seen in Table 3.1.

Parameter	Value [Unit]
Electric charge	108 [Ah]
Capacity	1630 [Wh]
Nominal voltage	15 [V]

Table 3.1: E-Golf, electrical specifications [10]



Figure 3.1: VW e-Golf battery module

Dimensions and mass of the BM can be seen in Table 3.2.

Parameter	Value [Unit]
Length	335 [mm]
Width	150 [mm]
Height	105 [mm]
Mass	11 [kg]

Table 3.2: E-Golf module dimensions and mass [10]

An exploded view of a BM can be seen in Figure 3.2, and is only used as a reference since it is not identical to the BM used in this project. Table 3.3 provides quantity, mass and material for the components of the BM provided by BatteriRetur AS. The mass provided in Table 3.3 are for each individual part. Table 3.3 will be further used in an disassembly analysis of the BM.

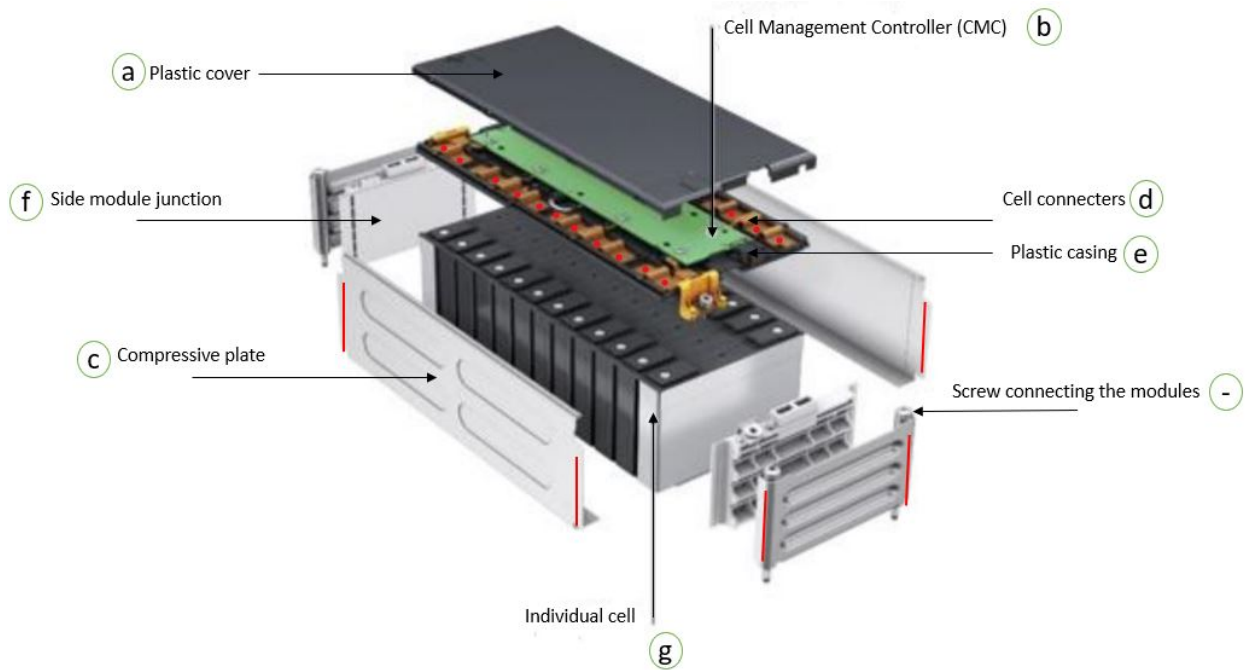


Figure 3.2: VW e-Golf battery module [1]

Parts name	Given letter	Qty.	Mass [g]	% of Total Mass	Material
Plastic cover	a	1	100	0.91%	Plastic
CMC	b	1	60	0.55%	Copper
Compressive plates	c	1	896	8.15%	Stainless steel
Cell connectors	d	8	N/A	N/A	Aluminium
Plastic casing	e	1	120	1.09%	Plastic
Side module junction	f	2	130	1.18%	Aluminium, plastic
Individual cell	g	12	797	7.25%	Aluminium

Table 3.3: Battery Module Components

3.2 Connection diagram

Manual disassembly of the module was conducted after discharging the module using electronic load. Fig. 3.3 shows how the module components are connected. The plastic cover (a) is fastened with snap-fit and a M4x8 TX20 screw, and the CMC(b) is fastened with six similar screws. The compressive plate (c) is laser welded with a lap joint indicated by the red lines in figure 3.2. The cell connectors (d) are also laser welded to the cells terminals, indicated by the red dots. The plastic casing (e) is a support structure for the CMC, and is held in place by the cell connectors. An example of manual disassembly, step by step, with the required tools can be seen in Table 3.4.

Step	Description	Required tool(s)
1	Removal of plastic cover (a)	Flat screwdriver and TX20 screwdriver
2	Remove CMC unit (b) with connectors	TX20 screwdriver, plier
3	Removal of compressive plates (c)	Grinder
4	Cut bridge between each cell connector (d)	Grinder or plier for cutting
5	Remove plastic casing (e)	Loose part, no tool required
6	Separate Side module junction (f)	Flat screwdriver

Table 3.4: Manual disassembly sequence

The most challenging fastening types are the ones which are irreversible, i.e. press fitted/mould, welds, rivets, adhesives [77]. These type of fastening methods are more time consuming when dismantling, and often require special tools. Therefore, a further research on the irreversible fastening methods utilized on the BM from Figure 3.1 is needed.

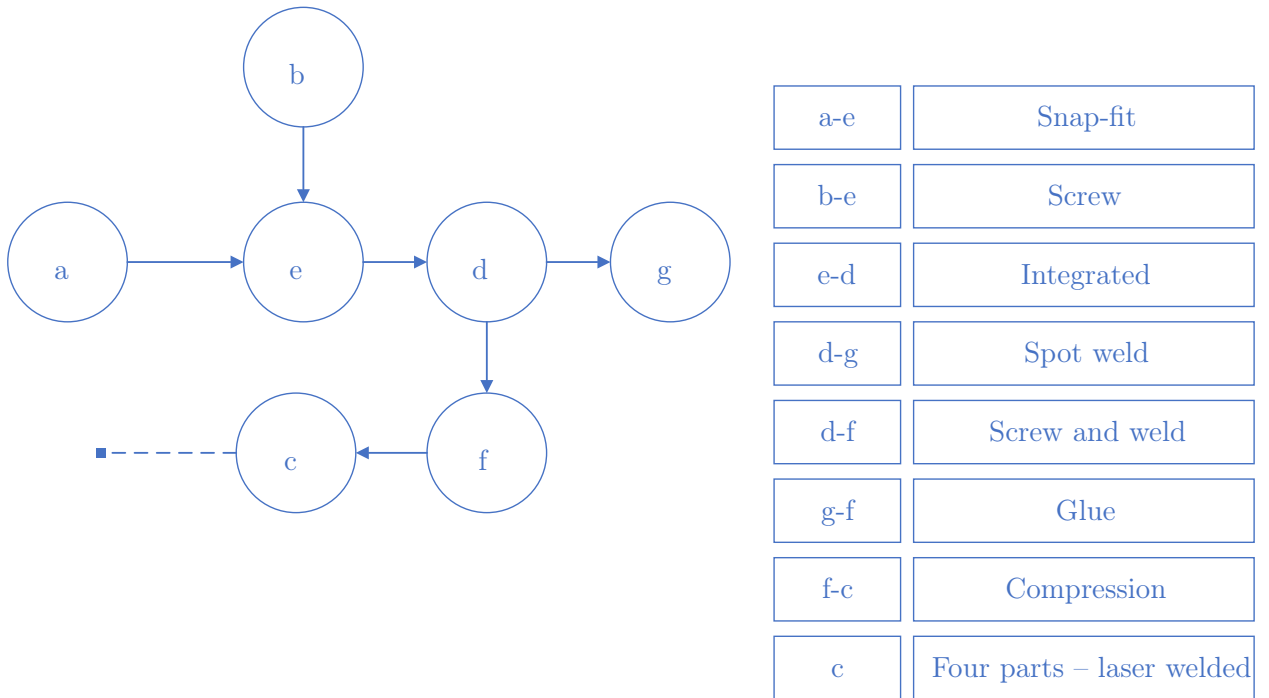


Figure 3.3: Connection Diagram of VW Ego1f-2019 BM

3.3 Laser Beam Welding

The connection diagram sec 3.2 shows that the compressive plates holding the cells in place, and the cell connector are laser welded using lap joint welding as seen in figure 3.4. Laser

Beam Welding (LBW) is broadly utilized in industrial application due to its numerous advantages such as controlled heating, low Heat Affected Zone (HAZ), deeper and narrower weld. HAZ is the area of the metal around the weld that is affected by the heat input shown in figure 3.5. Laser welding on stainless steel gives an ultimate tensile strength of 550 MPa according to [40], which is 71 MPa apart from the tensile strength of stainless steel which is 621 MPa. Another study [4] showed that laser welding technique does not weaken the material in the HAZ, due to its precision and controlled heating qualities which dictates that the weld (and not the HAZ) is the weakest link. Therefore, the laser welded samples fractured in the weld zone and not in the HAZ.

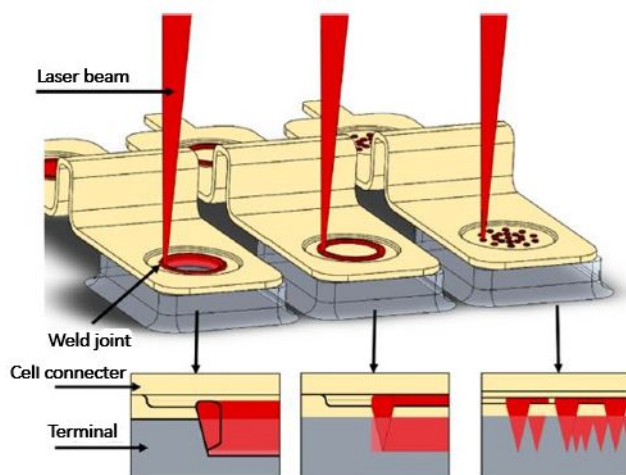


Figure 3.4: Standard methods of laser joining battery connections: fillet joint (left), lap joint (middle), multiple spot welds (right) [40]

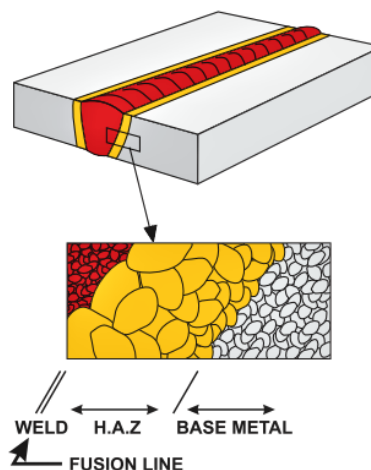


Figure 3.5: Heat Affected Zone (HAZ) [78]

Weld analysis

Dismantling techniques can be divided in three different categories [77]:

- Non-destructive
- Semi-destructive
- Destructive

Welding and soldering are categorized as irreversible fastening methods, and the preferred dismantling technique is either to saw/cut or break the weld/solder [77]. The laser welds connecting the compressive plates (c) motivates for a destructive dismantling technique. A solution with angle grinder is not desired, due to the dust from cutting, heat radiation (safety), and tool wear (operating costs). The remaining dismantling techniques that need further investigation are therefore the saw-/milling method, which would leave metal shavings instead of dust could be at an advantage and a hydraulic unit that could fracture the weld.

Weld dismantling techniques

A selection of dismantling techniques for the laser weld are listed in Table 3.5 with some of their advantages and disadvantages.

Technique	Advantages	Disadvantages
Angle Grinder	Fast, reliable	Heat, tool wear, dust
Saw	No dust (metal chips)	Slow, mixed with coolant,
Break	Fast, repetitive tasks	Cost

Table 3.5: Pros and Cons of different dismantling techniques

3.3.1 Tension test

To design a proper solution on how to disconnect the laser weld, further data characterising the weld strength are needed. Taking advantage of the available modules a strength test was performed in UiA laboratories. Combining a fixture and a tensile strength machine, it is possible to analyse the strength of the laser welds on the BM compressive plates. By fastening the BM compressive plates with two 8mm stainless steel rods through the sleeve where the screws connects the modules 3.2 it could be stretched apart until failure. From the experimental test, the required load to fracture the weld was 13.669 kN, see figure 3.7. The results from the experimental test can be seen in Figure 3.6.

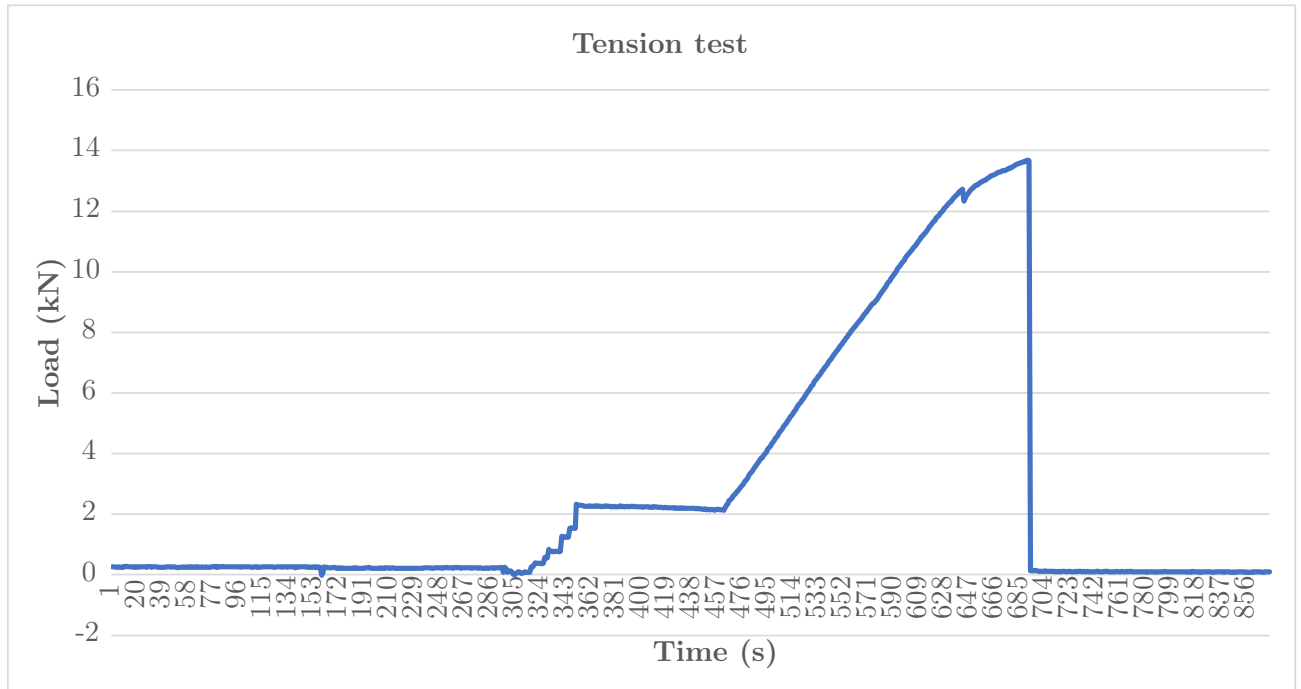


Figure 3.6: Tension Test Results, fracture at 13,669 kN

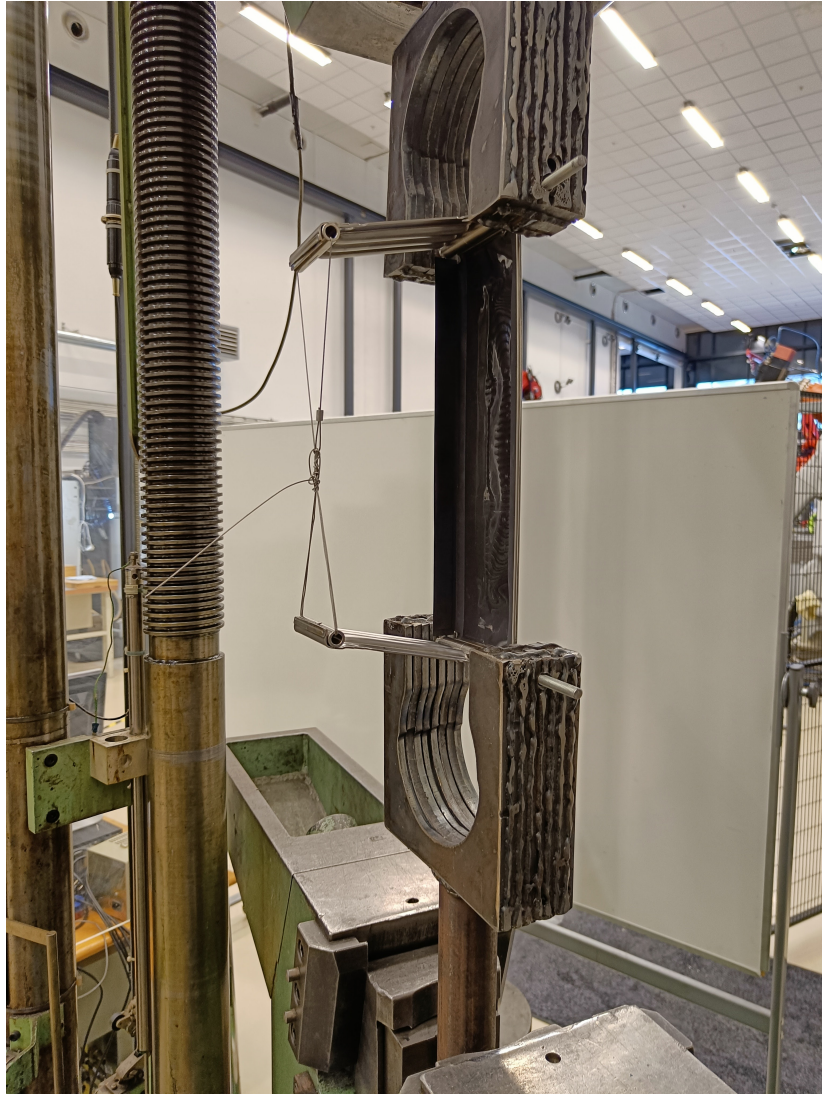


Figure 3.7: Tension Test Configuration

3.4 Dismantling Concepts

This chapter discuss and evaluate different dismantling concepts. Research, design and analysis of the concepts are presented.

3.4.1 Fracture of the laser weld

A concept using a hydraulic unit, i.e. consist of a hydraulic actuator and a fixture for the BM, is shown in Fig.3.8. The idea is to use a hydraulic actuator capable of generating the needed force to split the compressing plates apart, propagating a fracture in the weakest part: the weld.

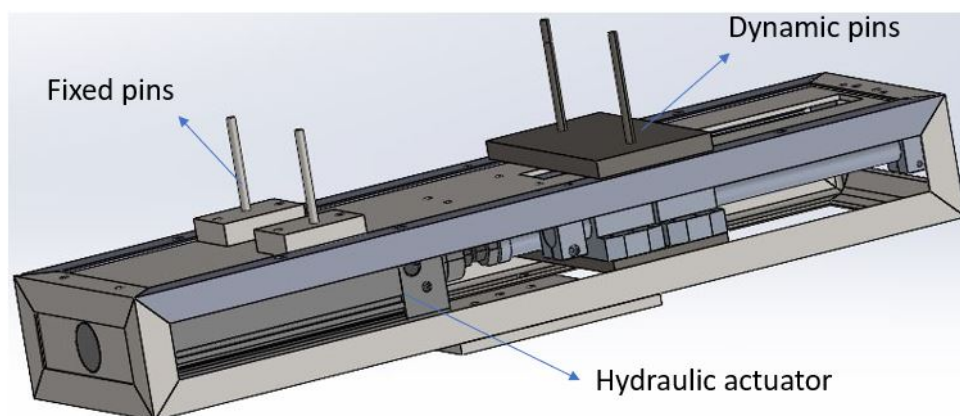


Figure 3.8: Hydraulic Pressure Unit

FEM-analysis:

Linear static FEM-analysis was conducted in SolidWorks to ensure a repetitive dismantling task which require strong pins that will not endure permanent deformation. SolidWorks was chosen rather than Abacus due to its user friendliness. The reason of choice is the modeling interface along with the flexibility for modifications of simulation properties and the ability of the software to troubleshoot and detect user input errors.

The tension test in 3.3.1 gave an indication of permanent deformation of the stainless steel pins. The goal of this analysis is to study how different configurations of the pins will preform under the given load, and if the pins will endure stresses greater than the yield strength of the material. The results from the tension test was used as an indication of the needed force to brake the case.

The first simulation was done with standard "DIN" steel, with the material properties listed in table 3.6. The load was applied as pressure distributed on half the surface area of the pin, eq. 3.1. Using force in the simulation is usually used when the applied load is concentrated to a specific point. Figure 3.9(a) shows the pins in two different configuration, were one of the pins is fixed in the top. The stress values is higher than the yield strength of the material in both cases which is not acceptable.

$$P = \frac{F}{A} \quad (3.1)$$

Further, a stronger steel type (1.2367) was used with the same study configuration. 1.2367 is a hot work steel which is usually used for wear resisting tools and pressure die casting tools. This material has higher yield strength as listed in table 3.6. The results seen in figure 3.10 shows that even this material would have difficulties withstanding the load and

the pins will endure permanent deformation. The pin that is fixed in the top had better chance in withstanding the load, showing that better structural design could improve the results. Unfortunately the diameter of the pins can not be changed to increase rigidness, since we are bound by the dimension of the sleeves. Taking into account the cost of the hydraulic unit using hot work steel for the pins, it would not be cost-effective to produce such a tool before more physical testing is conducted.

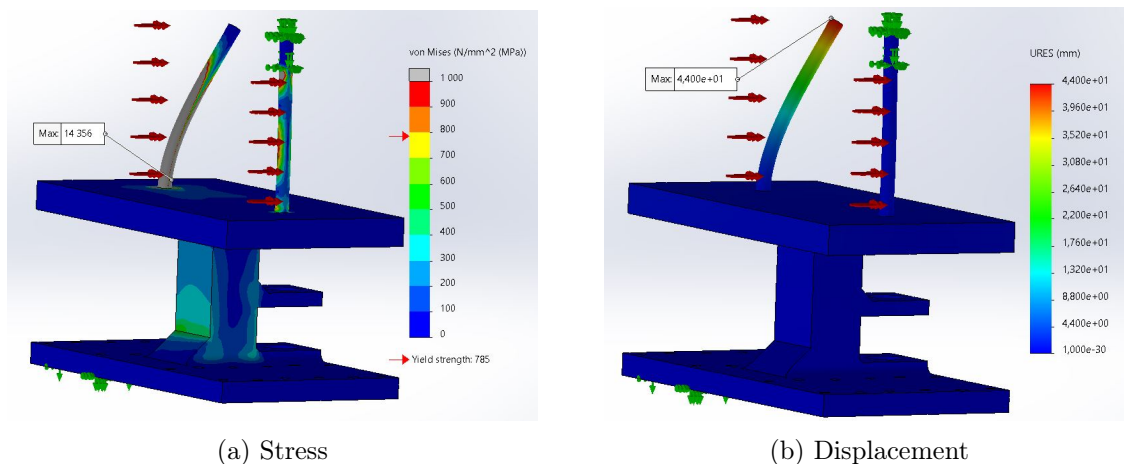


Figure 3.9: Standard Steel

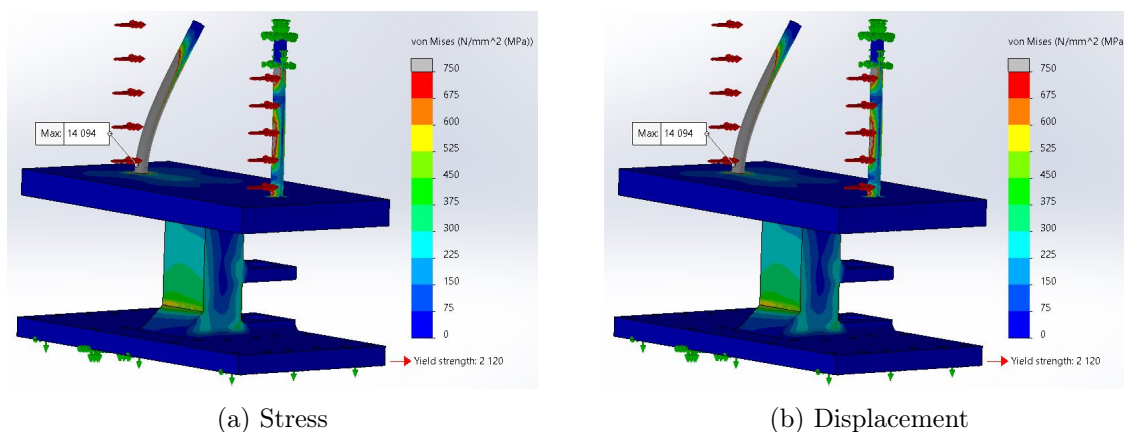


Figure 3.10: Hot Work Steel

Material	E-module [$\frac{N}{mm^2}$]	Yield strength [$\frac{N}{mm^2}$]	Tensile strength
Standard steel	215000	785	850
Hot Work Steel 1.2367	215000	2120	2120

Table 3.6: Material properties

Further, a representation of the welded compressive plates was modeled in SolidWorks and was assigned the assumed material properties (stainless steel). Stress accumulation in a specific point will indicate where the fracture will probably occur.

Simulating complex geometry in SolidWorks is time consuming and increases the probability of errors and inaccurate results. Due to complexity of the assembly when including the hydraulic pressure unit the study was conducted for the sole purpose of analysing the welded compressive plates. The study included four different configuration, long and short

pins with the load applied on both sleeves or only on one end of the module. This is done to analyse the effect of unsymmetrical load on the case. Materials behave differently when a load is applied in different direction, since the mechanical properties can be dependent on the direction of measurement.

Figure 3.11 shows how the sleeve was given a split line in the middle, which simulate short or long pins depending on which section of the inner surface that the load is applied to. The fixtures were also applied using the same method to simulate the contact surface of the sleeve and the pins. Table 3.7 describes in detail the properties of the simulation defined in SolidWorks. The computing time the solver uses is directly affected by the properties definition. The result of the analysis are presented in ch. 7.

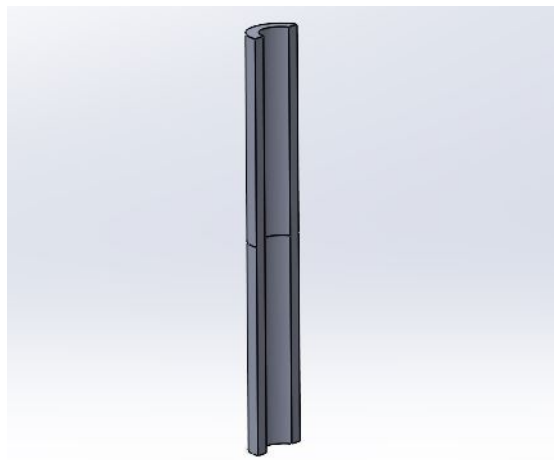


Figure 3.11: Section view of sleeve.

Simulation Properties	Description
Materials	<ul style="list-style-type: none"> Stainless steel sheet, assigned to the compressive plates E-module: 192999.9974, Tensile strength: 580, Yield strength: 172.37 [MPa] Stainless steel bar, assigned to the sleeves E-module 192999.9974, Tensile strength: 550, Yield strength: 137.89 [MPa]
Connections	Two contact sets of type "No penetration contact pairs" were defined in addition to the global contact. Each contact set prevent the sleeves from penetrating the compressive plates and transfer the applied load to the compressive plates. One contact set is defined for the sleeves where the load is applied, and one contact set for the fixed sleeves.
Fixtures	The fixtures used in the simulation are of the type "fixed geometry" and "roller/slider". The fixed geometry was applied to the sleeves on the opposite side of the load. The "roller/slider" type was applied to the loaded sleeves to ensure no vertical translation and to the long plates taking into account the battery cells enclosed in the casing, giving a more accurate and reasonable representation.
External loads	The load was applied as pressure on half the area of the inner surface of the sleeves, representing the contact surface between the sleeve and the pin.
Mesh	SolidWorks provides three types of mesh: solid, shell and breem mesh. Mixed mesh was automatically chosen since the study model represent different geometrical shapes. different parts of the model is given the suitable mesh. This allows a more accurate numerical results.

Table 3.7: Description of Simulation Properties

Hinged pins concept

Additionally, a proposed option is to attach the pins with a hing, as seen in Fig.3.12. This

will allow for a small rotation, thus having a torque that might induce a small fracture which will propagate further. This concept could not be tested safely due to the lack of equipment and fixture methods in the tensile strength machine.

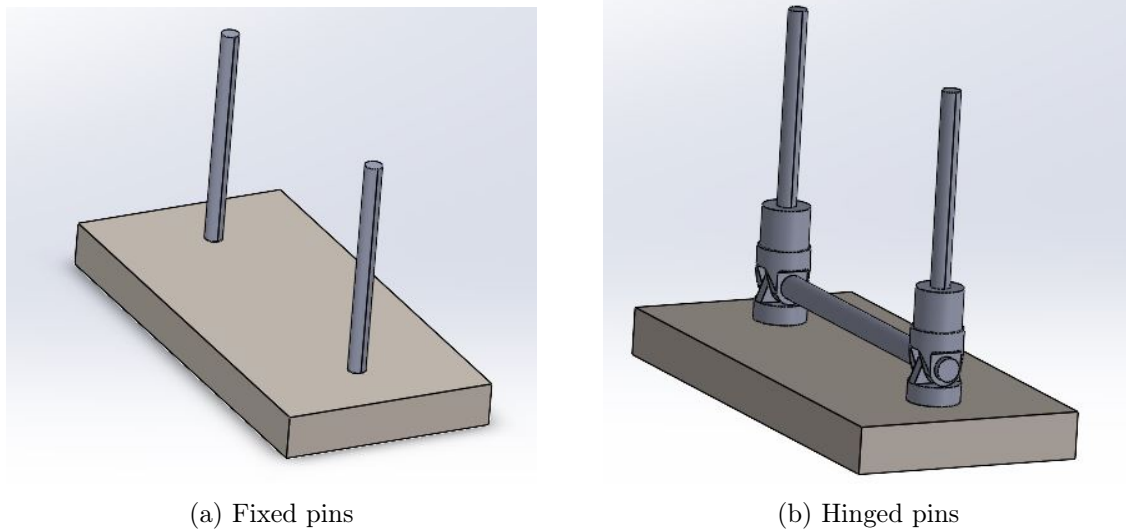


Figure 3.12: Fixed and Hinged pins concepts

3.4.2 Milling

A second concept to disconnect the weld is to combine traditional machining technique with advanced robotics. The utilization of robotic arms in milling operation has increased in the last few years. Integrated vision systems, force sensors and the advantages of the robotic arms large workspace are some of the factors that increased the interest in research and study of this field. Some of the gained advantages when using a robot arm for general milling operations are listed below.

- **Flexibility** The robot can reach the part in hand from multiple angles. It can mimic the features of a 5-axis CNC-machine with the right path planning tool.
- **Versatility** The robot can be used for multiple tasks depending on the end effector installed, while a CNC-machine is assigned with only one task.
- **Cost efficiency** CNC-machines and robots are expensive, but the versatility of an industrial arm will make an impact on the economic aspect of any production line.
- **Accuracy** The highest achieved accuracy of a CNC-machine is $\pm 2\mu m$ while KUKA KR210 has achieved an accuracy of up to $200\mu m$. The accuracy would weigh more if the application depends on this level of high accuracy, which is not the case in the dismantling task.

According to [81] the stiffness of a robot arm in general is 50 times lower than a CNC-machine, which leads to higher position error during a milling operation. Initially, the main focus for robot arm accuracy has been gravity, load and deflection compensation. Currently, due to research on robot-stiffens and real-time force sensing the problem of position error and deflection has been solved. Combining a stiffness model in the robot controller has increased the accuracy of robot arm milling operations. Its is necessary to utilize interactive-force compensation for the purpose of attaining higher accuracy.

The main application of the ABB IRB4400 is cutting, deburring, polishing and measuring. A stronger robot with higher stiffness should be used for milling the side of the battery

module. Several robot models from different manufacturers can be used for this purpose including ABB IRB6600 and ABB IRB6660.

ABB IRB6600

In [71] ABB IRB6600 seen in Fig.3.13 was used to test off-line path correction of robotic face milling using static tool force and robot stiffness. This paper showed that it is possible to compensate for the deflections of the tool when milling even without a real-time force/torque sensor feedback, which can be a cost-effective method. This was done by estimating the robots joint stiffness experimentally, in addition a process force estimation model. The force estimation model counteract the interactive force between the tool and the work piece.



Figure 3.13: ABB IRB6600 in Mechatronics Lab at UiA

ABB IRB6660

ABB IRB6660 is optimized for high performance applications such as milling, cutting and sawing. This model is heavier, have greater handling capacity and longer reach than the 4400 series. The robust design give the robot body the needed stiffness for milling operations. This robot comes with a special protection kit. It consist of different enhancement of the different parts of the robot for improved protection. The motors and connectors has special seals and covers for environment adaptation and the whole robot is painted with two-component epoxy.

This IRB6660 is available in three different versions with different load handling capacity and reach. All three versions are capable of machining and should be evaluated according to the design requirements of the robotic cell. Table 3.8 gives a comparison of the different versions of the IRB6660, note that the power consumption is measured with maximum load while the robot is moving in a ISO Cube at maximum velocity.

Stainless steel milling:

When choosing the proper tool for milling, the characteristics of the tool are depending on stainless steel properties. There are two main issues when milling such a hard material, strain hardening and large cutting force. Strain hardening occurs when extreme stresses

Version	Load capacity [kg]	Reach [m]	Weight [kg]	Power Consumption [kW]
IRB 6660-130/3.1	130 kg	3.1 m	1910 kg	2.3 kW
IRB 6660-100/3.3	100 kg	3.3 m	1950 kg	3.1 kW
IRB 6660-205/1.9	205 kg	1.9 m	1730 kg	3.6 kW

Table 3.8: ABB IRB6600 version comparison

causes metals to harden, which increases the wear of the cutting tool. The cutting force increases due to the plastic deformations of stainless steel during cutting. High cutting speed (V_c) should be applied to overcome the strain hardening effect. The tool's material should be extremely strong and heat resistant, tungsten-cobalt alloys are commonly used. The tool geometry is also taken into consideration, the rake angle should be large to minimize the increase in cutting force caused by deformations.

A manual milling operation was conducted in order to examine the milling concept for the dismantling task. A suitable tool seen in Fig. 3.14 was chosen for the test with the following specifications:

- **Brand:** Dormer
- **Type:** End mills
- **Overall length:** 83 mm
- **Length of cut(maximal cut depth):** 26 mm
- **Diameter:** 12 mm
- **Rake angle:** 25°

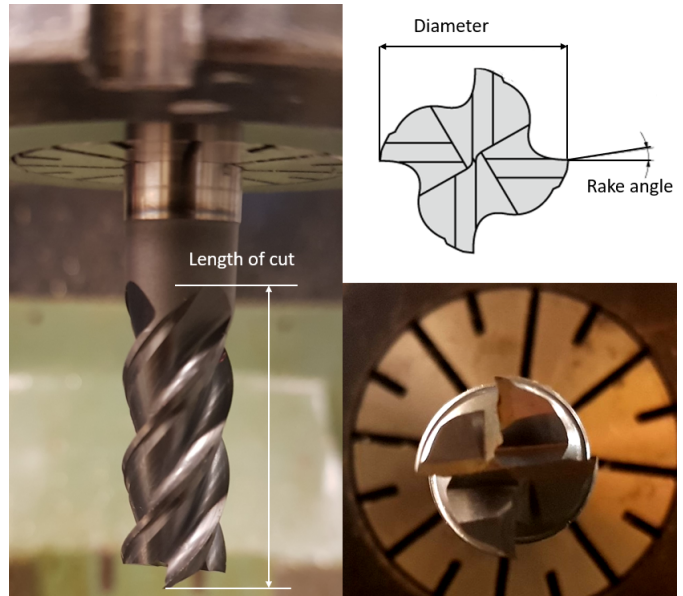


Figure 3.14: Stainless Steel Milling Tool

The tool was selected to cover the entirety of the laser weld, thus a diameter of 12 mm. Further, the module was fixed in a two-jaw chuck on the table and the tool was centered right above the weld. On the short side of the module the compressive plates are thicker and they are bent 90° around the sleeve. The laser weld on the long side connects the short and long plate together, while on the short side the weld connects the plate with the sleeve. Hence, the detection of the sleeves will aid in path planning for the end effector.

Fig. 3.15 shows the coordinate frame of the tool end point when it was centered before initiating the cut, the feed in Y-direction and the module after the milling operation. The thickness of the long plate is 0.6 mm, the tool was lowered 0.8 mm in the Z-direction for the initial cut considering the weld penetration depth. Manually the Y-axis handle was turned until the cut was complete. Since the plates are compressed before welding, the plates snap and are effortlessly dismantled.

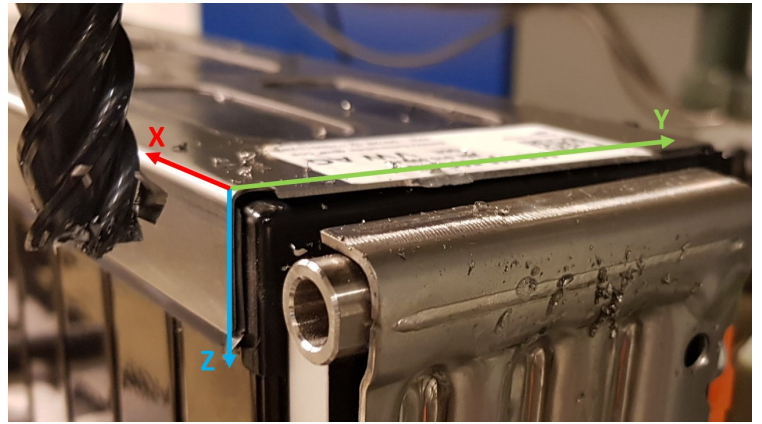


Figure 3.15: Stainless Steel Milling Tool

Chapter 4

Experimental setup

This chapter presents the hardware used in this project. A proposed system architecture for how each equipment should be interacted with respect to software is presented. Experimental setup also includes a guide/workflow on how to configure the robotic setup on a laptop or desktop.

4.1 ABB IRB 4400/60 + ABB IRBT 4004

The robotic setup in this project is a combination of a ABB IRB 4400 (/60) manipulator mounted on a ABB IRBT 4004 track ,increasing the manipulators working area. The IRB 4400 is a 6-axis manipulator that was created with the manufacturing industry in mind. The robot features built-in process ware, an open structure that is expressly designed for flexibility, and can interact with external systems effectively. This manipulator has a load capacity of $60kg$, and a maximum gripping reach of $1.96m$ [3]. Dimensions of ABB IRB4400 can be seen in Fig. 4.1. A *single cabinet ABB IRC5 controller* [2] is utilized with the

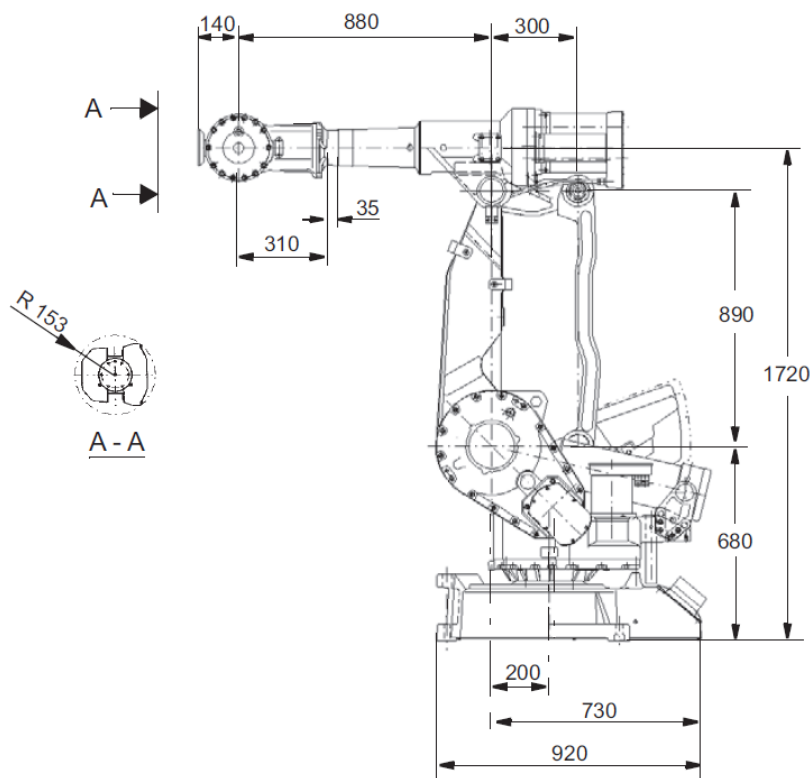


Figure 4.1: ABB IRB4400 Dimensions [3]

IRB4400 and IRBT4004. By connecting the controller with ROS, it is possible to feed joint

inputs for the robotic setup to achieve inverse kinematic control with ABB ROS driver, and MoveIt.

4.2 Zivid One+ Medium

Zivid develop color cameras for industrial robots with 3D machine vision. Zivid One+ Medium was utilized for this project. Zivid One+ is a RGB-D camera which combines 2D images with the 3D technology, structured light. The optimal working distance is $700 - 1500[mm]$, and a focus distance of $1000[mm]$ for Zivid One+ M's. Images is captured in the 1920×1200 resolution, i.e. 2.3Mpixel [52]. Meaning each capture generates 2,3 million pixels in the image frame. Zivid also provides a wrapper to be used with ROS. More about the Zivid-ROS wrapper is presented in 5.1. As an eye-in-hand configuration, the Zivid camera was mounted on the manipulators tool-center-point (tcp).

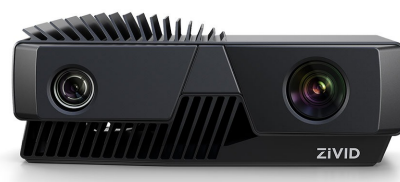


Figure 4.2: Zivid One+ camera

4.3 Computer hardware

Some computer operations require a minimum of computational power from the hardware. As for the Zivid SDK, they recommend a minimal compute capability of 3.3 for a dedicated Graphics Processing Unit (GPU) [85]. Compute capability of Nvidia's graphics cards can be found on their developer page [48]. The laptop utilized for computer vision and robotic control is a Lenovo Legion 5. Specifications of the project laptop which affects computational performance can be seen in Table 4.1.

Component	Type
Central Processing Unit (CPU)	AMD Ryzen 7 5800H (8 Cores)
GPU	NVIDIA GeForce RTX 3070 Mobile
Random Accessible Memory (RAM)	16GB (3200MHz)
Storage	M.2 SSD 1 TB

Table 4.1: Specifications project laptop

A graphics driver is mandatory to utilize the fully potential of a dedicated graphics card.

4.4 System architecture

The proposed system architecture can be seen in Fig. 4.3. By combining ROS packages like MoveIt, and ROS-wrapper for Zivid. Allows the user to maneuver the manipulator, and capture images from a laptop or computer running Ubuntu.

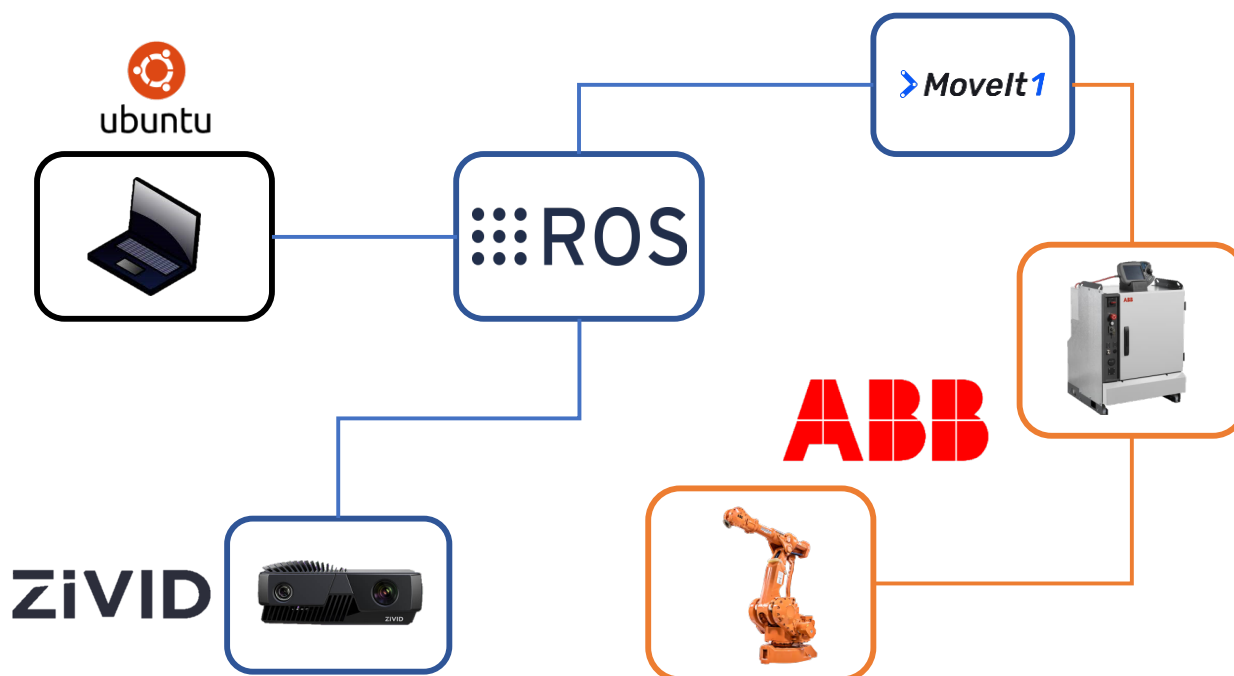


Figure 4.3: Flow chart, proposed architecture

4.5 UiA Robotics lab setup

This section will present setup and utilization of the UiA mechatronics lab (lab) with a laptop. It was also intended to be used as a guide for future projects at the lab.

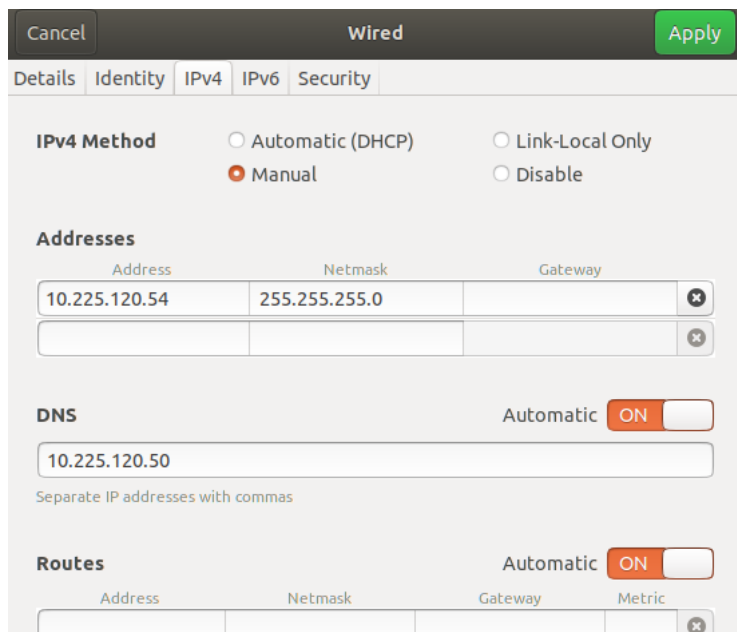
The lab was originally setup with a distributed ROS structure. Two stationary computers with Ubuntu 16.04 and ROS, one as a master (server/rack pc) where the ROS core is running. The other computer is communicating with the ROS core (rack pc) as a node. For the purpose of this project, it was desired to build a system which could be utilized using a laptop. The lab computer setup runs ROS Kinetic with Ubuntu 16.04 serving the core as a node. As for today, ROS Kinetic is considered as an EOL product, and packages for ROS kinetic are no longer developed [59]. Therefore ROS Melodic was a preferable candidate, since the driver for ABB IRC5 controllers (*abb_driver*) was developed with respect to ROS Kinetic and ROS Melodic.

Software installation

The recommended Ubuntu OS version for ROS Melodic is Ubuntu 18.04 [62], follow [Ubuntu's guide](#) for installation utilizing a USB flash drive. An Ubuntu install could be performed by creating a partition in your existing OS (e.g. Windows), or by installing it on a clean storage unit (HDD/SSD) which was used in this project. ROS provides a step-by-step guide on how to install [ROS Melodic](#), the *desktop full* version is recommended. All dependencies for package building in ROS utilize *Python 2.7*.

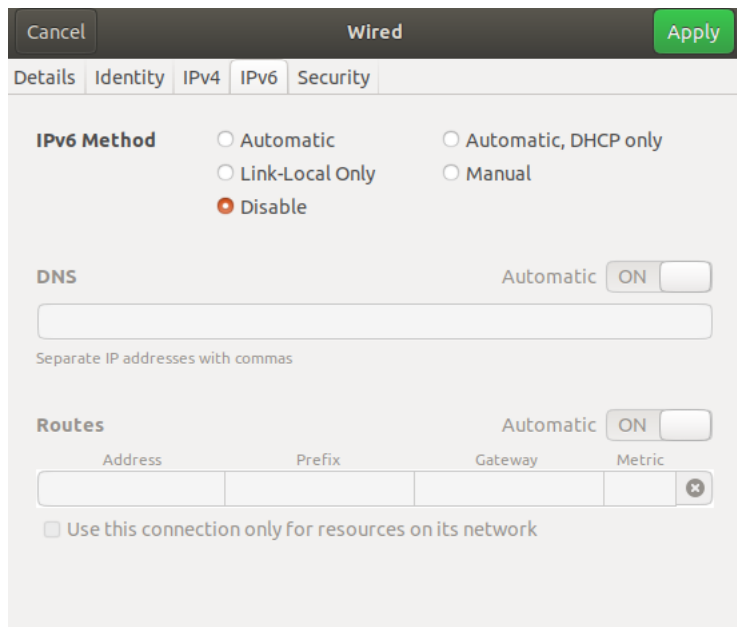
Network- and .bashrc file configuration

Network settings on the computer or laptop has to be modified in order to establish connection to the robot. First, connect the Ethernet cable to the laptop or computer. Then go to *Network settings > Edit connections > Select Ethernet > Add > Choose Ethernet as the connection type*. In the IPv4 settings tab, set the Method to *Manual*, and the addresses should be corresponding to Figure 4.4. Under the tab *Identity*, the *Name* can be configured by choice. In the IPv6 tab, set the Method to *Disable* as in Figure 4.5. Keep in mind, these configurations are with respect to the "righty" robot in the lab.



The screenshot shows the 'Wired' network settings window with the 'IPv4' tab selected. The 'IPv4 Method' is set to 'Manual'. The 'Addresses' section contains two rows: the first row has '10.225.120.54' for Address, '255.255.255.0' for Netmask, and an empty field for Gateway; the second row is empty. The 'DNS' section has 'Automatic' set to 'ON' and a text field containing '10.225.120.50'. The 'Routes' section has 'Automatic' set to 'ON' and an empty table with columns for Address, Netmask, Gateway, and Metric.

Figure 4.4: IPv4 settings



The screenshot shows the 'Wired' network settings window with the 'IPv6' tab selected. The 'IPv6 Method' is set to 'Disable'. The 'DNS' section has 'Automatic' set to 'ON' and an empty text field. The 'Routes' section has 'Automatic' set to 'ON' and an empty table with columns for Address, Prefix, Gateway, and Metric. There is also a checkbox labeled 'Use this connection only for resources on its network' which is currently unchecked.

Figure 4.5: IPv6 settings

In order to establish connection between ROS and the robot (righty), the `.bashrc` file must be modified. The `.bashrc` file is executed when the computer starts up as a script. It contains series of configurations for the terminal session. It includes setup or enabling of some of the following [33]:

- Coloring
- Completion
- Shell history
- Command aliases

The `.bashrc` file can be located in the *Home tab* under *Files*, and check *Show hidden files* drop down menu as shown in Figure 4.6. You can also access it from the shell terminal with the following command:

```
~$ sudo nano .bashrc
```

The ROS hostname, and ROS IP must be configured at the end of `.bashrc` as shown below.

```
#Define robot IP
export ROS_IP = 10.225.120.54
#Define HOSTNAME
export ROS_HOSTNAME = #your name#
```

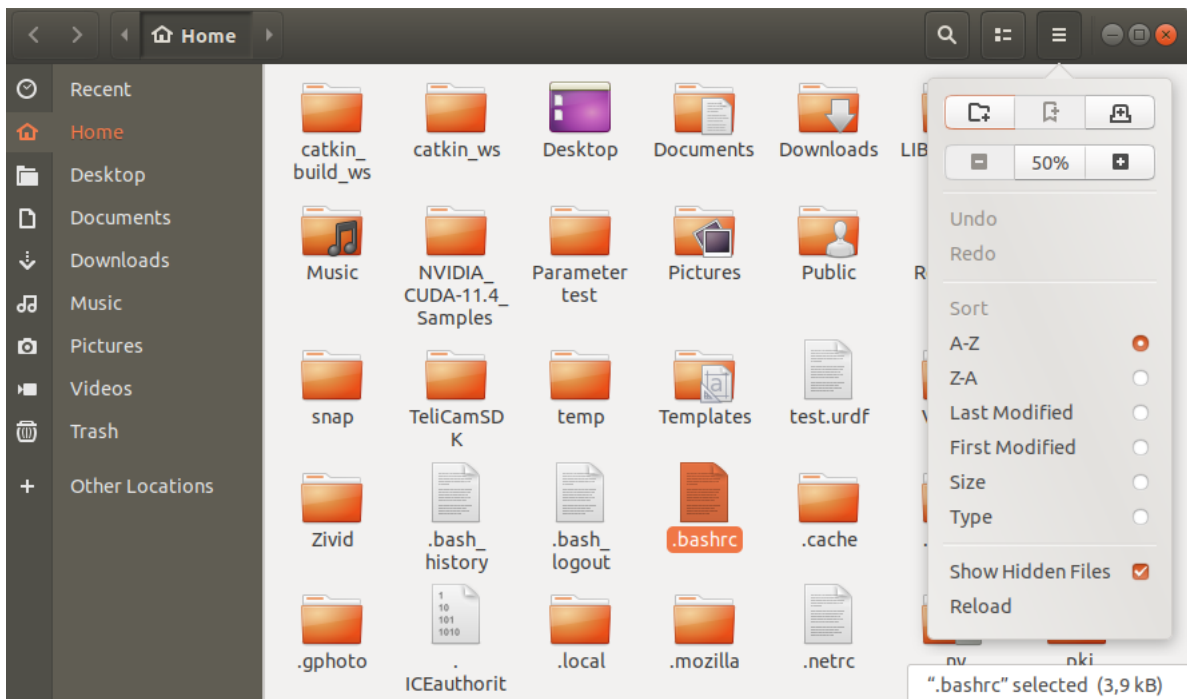


Figure 4.6: `.bashrc` file location

Catkin Workspace

After ROS installation, network- and `.bashrc` configuration, a `catkin workspace` must be configured. *NOTE: Always remember to check that you are using guides corresponding to the specific ROS distribution used.* SFI Mechatronics developed ABB ROS support (including driver) for the robots in the lab, and can be found at their [Github](#) page. Clone this repository into the `src` folder in the catkin workspace, and then rebuild. This is done in the terminal:

```
~$ cd catkin_ws/src
~$ git clone https://github.com/SFI-Mechatronics/wp3_abb.git
~$ cd ..
~$ catkin build
```

Simulation

wp3_robots-master is a folder containing *MoveIt!* configuration packages for the *lefty*- and *righty* robots, located on the lab computer. These folders includes configuration- and launch files for each robot. Compress this folder as a *.zip*, then extract it in the *src* folder in the *catkin workspace* on the computer to be used for your project. Then rebuild the *catkin workspace*.

```
~$ cd catkin_ws
~$ catkin build
```

In order to work with the *MoveIt!* configuration packages, *MoveIt!* must be **installed**. Within the *launch* folder, there is a *demo.launch* as seen in Figure 4.7. *demo.launch* brings up the a 3D-model of the *righty* robot (ABB IRB 4400 + ABB IRBT 4004) in RViz as seen in Figure 4.8. The robot can be simulated by clicking on the spherical ball and move it to a coordinate, or with the *move group python interface* [38] from *MoveIt!*. It's recommended to go through their tutorials, if you have not worked with *MoveIt!* before. A script where the python interface was tested can be seen in Appendix D. The *demo.launch* can be launched with the following command in the terminal:

```
~$ roslaunch righty_moveit_config demo.launch
```

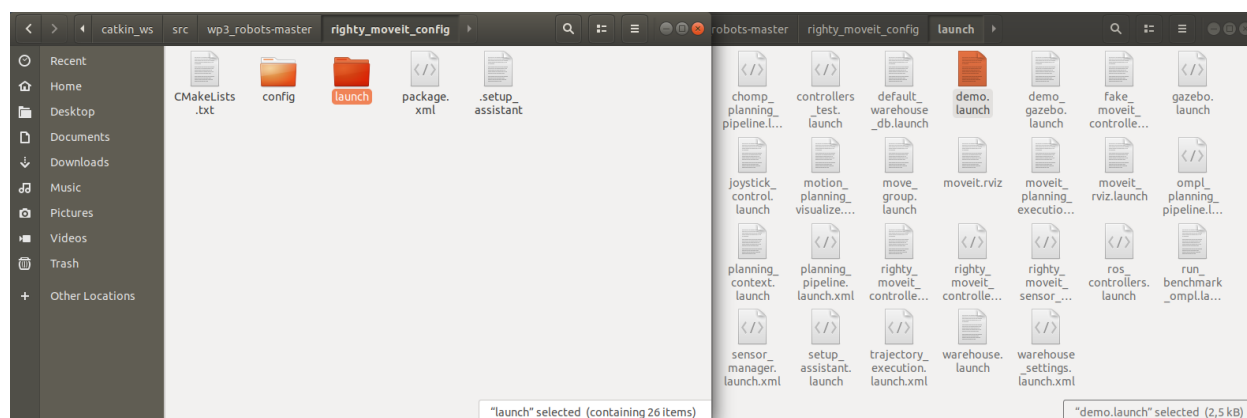


Figure 4.7: *demo.launch*

It is common to get errors for unreachable positions when defining a pose goal. A solution could be to change your motion planner (i.e RRT, RRTConnect, RRTStar...). A helpful tool is to add axis of the tool center point (tcp), by default this is *tool1*. In RViz, click *Add > Axes > OK*. As for *reference frame*, select *tool1*. Another tool which could be helpful, is a 3D rotation converter [41]. It converts Euler angles (either in degrees or radians) into quaternions, or vice versa. It can make it more understandable to reach your *pose goal orientation* as desired. Coordinates for your pose goal position are with respect to the world frame configured in the *MoveIt!* configuration for the robot. The world frame is located in the corner of the lab as shown in Figure 4.9. In order to make your script executable, run the following command in the terminal:

```
~$ cd *folder_name*
~$ chmod +x *file_name*.py
```

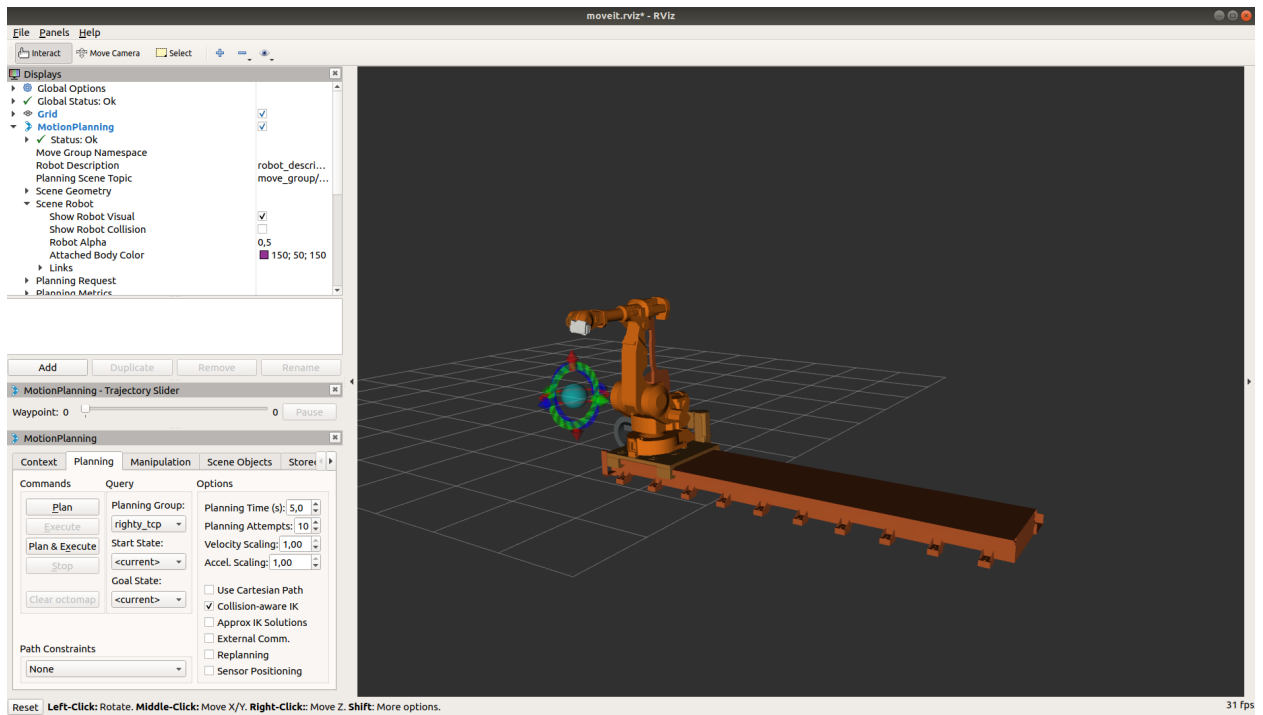


Figure 4.8: 3D visualization in RViz

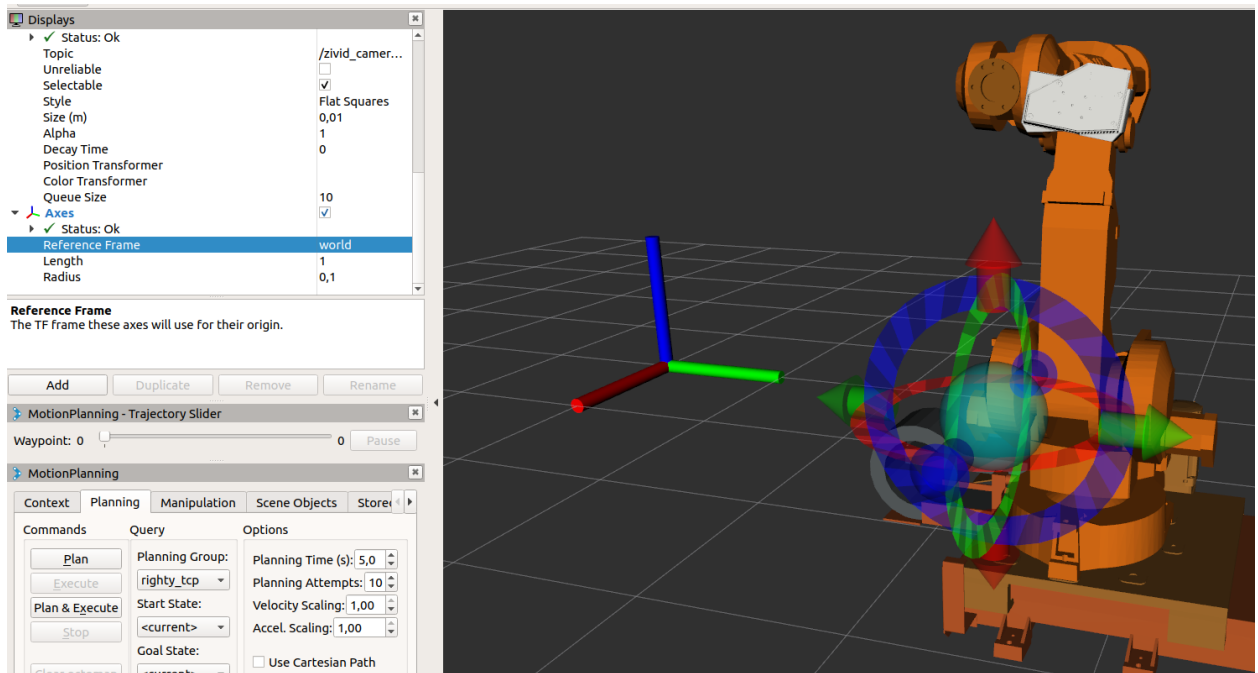


Figure 4.9: World frame in RViz

Control of the robot

If the pose goal positions and -orientations seems correct in the simulation, the next step would be to control the robot psychically. The robot can be simulated with the controller displaying the real-time joint states by running the following command:

```
~$ roslaunch wp3_robots show_righty.launch
```

Before working with the robot, the revolution counters must be updated. On the teach pendant go to *Calibration* > *ROB_1* > *Update revolution counter* > *Yes (Warning screen)* > *Check the marker box for the track and robot* > *Press OK* > *Update*. In order to connect

with the robot, the controller must be set up to get inputs from ROS. At the ABB main menu on the teach pendant, go to *Program Editor > Debug > PP to Routine > ROS_main*. Then the .launch file can be executed in the terminal as shown below:

```
~$ roslaunch wp3_robots load_robot_mod.launch
```

When launching, the info screen on the pendant will confirm connection, "*Client at 10.225.120.54 connected*".

Chapter 5

Image acquisition

The use of image processing as a tool for non-intrusive precision measurement, autonomous robot navigation, and dependable verification of industrial automation processes is growing rapidly in the engineering community. For a long time, physics, computer, electrical, and mechanical engineering sciences have been familiar with image acquisition and processing techniques [35]. This chapter presents the adjustment of Zivid's adjustable parameters to achieve the best result for the object detection.

Zivid Camera has an official ROS driver. The driver supports Ubuntu version 16.04 (ROS Kinetic), 18.04 (ROS Melodic) and 20.4 (ROS Noetic). ROS must be installed including configuration of a catkin workspace as mentioned in Chapter 4.5. Zivid provides step by step installation of the driver and essential software such as Zivid Core on the Github page [84]. It is necessary to have an external graphics processing unit (GPU) with a corresponding GPU driver.

In this chapter, acquisition settings for the Zivid RGBD camera have been tested and evaluated by utilizing the Zivid ROS driver.

5.1 Zivid ROS driver

Zivid provides sample captures and tutorials for the Zivid camera. Both with *C++* and *Python*. For this project, *Python* has been utilized. There is services which can be utilized included in this driver, and data published to multiple topics.

Services

The included services for the Zivid ROS driver are described below.

- **capture_assistant/suggest_settings:**
zivid_camera/CaptureAssistantSuggestSettings.srv
This service provides suggested settings for the camera, by analysing the scene with ambient light, distance, etc. as input. The capture assistant has two parameters:
 - **max_capture_time (duration):** Specify the capture time for each capture. 10 seconds is the maximum capture time, while 0.2 seconds is the minimum. A longer capture time can provide better data for challenging scenes.
 - **ambient_light_frequency (uint8):** Possible input values are either *50Hz*, *60Hz* or *NONE*. For optimal performance, used input *NONE* if ambient light is unproblematic.
- **capture:**
zivid_camera/Capture.srv
Service which triggers a 3D capture. After this service is settled, data is published on topics over ROS. These topics are described in subsection 5.1.

- **capture_2d:**
zivid_camera/Capture2D.srv
Similar to *capture*, however this service only triggers 2d capture where the image is published on the same topic as mentioned above.
- **load_settings_from_file & load_settings_2d_from_file:**
zivid_camera/LoadSettingsFromFile.srv & *zivid_camera/LoadSettings2DFromFile.srv*
In Zivid's software developer kit (SDK), Zivid Studio, acquisition settings can be exported as an *.yaml* file. Visible via *dynamic_reconfigure*. Settings can be utilized when triggering *capture*- or *capture_2d* service.
- **Camera_info/model_name**
zivid_camera/CameraInfoModelName.srv
When provoked, this service returns the model name of the camera.
- **Camera_info/serial_number**
zivid_camera/CameraInfoSerialNumber.srv
When provoked, this service returns the serial number of the camera.
- **is_connected**
zivid_camera/IsConnected.srv
If provoked, this service returns a message displaying if the camera is in connected state or not, with respect to ROS. It is updated before each capture, and automatically every 10 seconds. It will attempt to reconnect, if it's not in the connected state.

Topics

Topics such as 3D, color, signal to noise ratio (SNR), and camera calibration are provided by the Zivid ROS driver as a result of calling the capture services. Some of the topics are listed below and can be utilized for different use cases.

- **color/camera_info**
sensor_msgs/CameraInfo
Calibration of the camera as well as metadata.
- **color/image_color**
sensor_msgs/Image
By calling the *capture*- or *capture_2D* service, color-/RBGA image is published on this topic. Encoded as "rgba8".
- **depth/image**
sensor_msgs/Image
Topic containing depth image. a Z-value [m] is stored in each pixel along the camera Z axis.
- **points/xyzrgba**
sensor_msgs/PointCloud2
Point cloud data is published to this topic, when capture service is triggered. Output in camera's optical frame. X (right), y (down), z(forward), published in meters.
- **points/xyz**
sensor_msgs/PointCloud2
Similar to topic *points/xyzrgba*, however XYZ 3D coordinates are included.
- **snr/camera_info**
sensor_msgs/CameraInfo
Metadata and camera calibration

- **snr/Image**
sensor_msgs/Image
Signal-to-noise ratio data included in each pixel.
- **normals/xyz**
sensor_msgs/PointCloud2
Point Cloud normals. Included fields are normal x,y and z coordinates. Float value for each coordinate, and the normals are unit vectors. Subscribing to this topic will cause extra processing time.

5.2 Acquisition settings

When capturing images with the camera from Zivid, it is possible to specify desired acquisition settings. *settings/acquisition_<n>/* contains settings for an individual acquisition. *<n>*, can be defined for 10 total acquisitions, from acquisition 0 to 9. The parameters which can be specified for each acquisition can be seen in Table 5.1

Name	Type	Zivid API Setting
<i>../acquisition_<n>/enabled</i>	bool	
<i>../acquisition_<n>/aperture</i>	double	Settings::Acquisition::Aperture
<i>../acquisition_<n>/brightness</i>	double	Settings::Acquisition::Brightness
<i>../acquisition_<n>/exposure_time</i>	int	Settings::Acquisition::ExposureTime [μ s]
<i>../acquisition_<n>/gain</i>	double	Settings::Acquisition::Gain

Table 5.1: Acquisition parameters

Aperture

Aperture or Iris aperture is a definition of the light-transmitting ability for a lens. Iris diaphragms are usually fitted in lenses calibrated in units of relative aperture, a number of N . The Relative aperture is defined as the equivalent focal length f , divided by the entrance pupil diameters of the lens d for infinite focus. See Equation 5.1.

$$N = f/d \tag{5.1}$$

If the entrance pupil diameter is e.g. 25mm, and the focal length 50mm the relative aperture would be 50/25, i.e. 2. Usually, the numerical value of the relative aperture is prefixed with the italic letter f , and an oblique stroke, e.g. $f/2$. The denominator in the expression is often referred to as the *f-number* of the lens. *Aperture* or *f-stop* can also be referred to as the relative aperture. The largest diaphragm opening which can be used with the lens, corresponds with the aperture relative to the maximum aperture. From a standard series of number, *f-numbers* are selected to simplify exposure calculations. Each of the standardized numbers is related to the next by a constant factor. The factor is chosen due as the amount of light passed by a lens is inversely proportional to the square of the *f-number*. Each number in the standardized series is increased by a factor of $\sqrt{2}$, i.e. 1.4. Therefore, the standardized number are: 1, 1.4, 2, 2.8, 4, 5.6, 8, 11, 16, 22, 32, 45 and 64. A lens may have *f-numbers* between the standardized numbers, then to be marked with a number not in the standard series [23].

The Zivid camera has integrated electro-mechanical iris which can be adjusted rapidly. A *f-number* of $\frac{f}{1.4}$ corresponds to a completely open lens, while *f-number* of $\frac{f}{32}$ is considered a closed lens. Relative aperture is co-related to the depth of field, meaning the image range which is in acceptable sharp focus for a chosen aperture. A narrow depth of field (short focus length) would be the result of a relatively low *f-number*/wide aperture. Thus, a greater depth

of field would be the result of a high *f-number*/narrow aperture [86]. Figure 5.1 shows the co-relation of the depth of field [mm] and *f-number* for the Zivid One+ Medium camera.

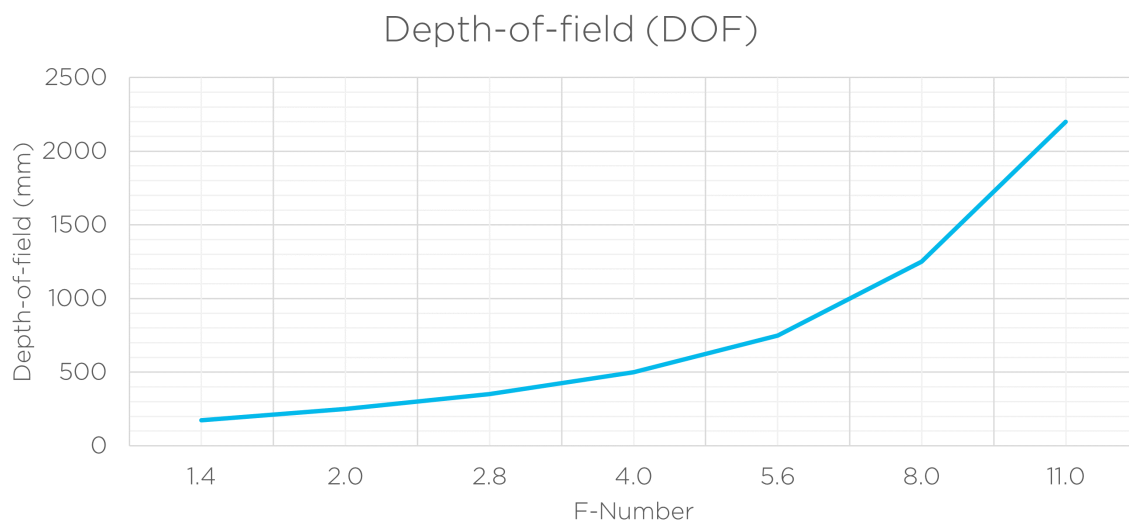


Figure 5.1: Depth of field, Zivid One+ Medium [86]

Stops

"Stops", also known as "Exposure stops" is an expression used to describe the level of brightness of an image, or how much light which hits the lens/sensor. This is relative to a reference level, which occasionally is referred to as "0 stops". Moving one stop up, while double the brightness of an image, while reducing one stop will reduce the image brightness by half. The sum of how many times the light intensity of a camera can be doubled is correspondent to the amount of stops a camera has available. The Zivid camera has 23 stops approximately, which is why Zivid provides superior data on shiny objects [86].

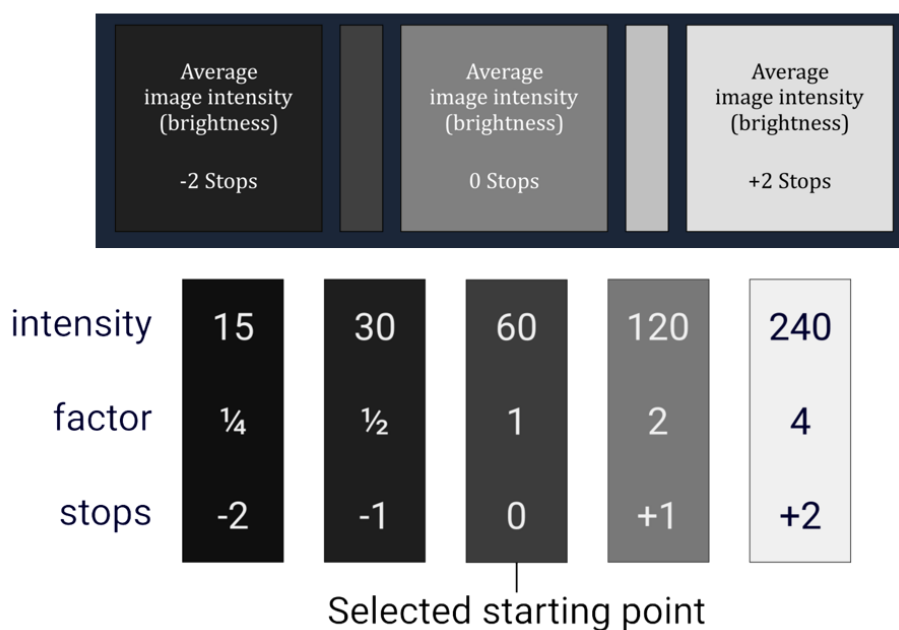


Figure 5.2: Stops in practice [86]

Introducing *stops* in practice, as shown in Figure 5.2. Imagine a default brightness or average image intensity value of 60 when referring to "0 stops". By moving 1 stop up, the brightness

is doubled to 120. Moving another stop up, the brightness value is 240. Also, when moving 1 stops down from the default value, the average image intensity will be reduced by half from 60 to 30.

Brightness

The luminance of objects is linked to the visual sensation we refer to as brightness [27]. An Image will be brighter for the viewer, the higher luminance of the screen. Image size and reflective properties of the screen, are factors which must be accounted when referring to the luminance output of the screen. The luminance L perceived by a viewer is dependent of a screen gain $G(\theta)$ which describes reflection of light by a screen in a specific direction θ . In addition to the screen gain, image area A and projector brightness Φ must be included for calculation as seen in Equation 5.2.

$$L = \frac{G * \Phi}{\pi * A} \quad (5.2)$$

The parameter brightness, or projector brightness controls the amount of light emitted from the projector by adjusting the power output. The most efficient way to maximize the signal-to-noise ratio (SNR), is by manipulating the projector brightness. In order to maximize the amplitude of the signal received by the camera, the projector brightness should be maximized. This would result in a minimized impact from noise in the signal, as long as the projector does not over-saturate the pixel from the reflected light. The mean intensity of the image is affected by increased projector brightness, also minimized noise impact which can be seen in Figure 5.3. Meaning the exposure, measured in stops can be controlled with the projector brightness [86].

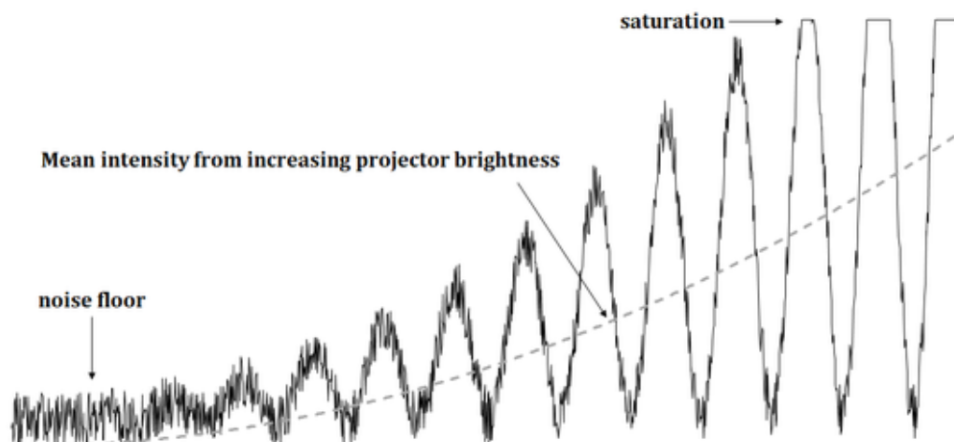


Figure 5.3: Mean intensity from increasing projector brightness [86]

The default value for the projector brightness parameter is 1.0, and is based on a relative value. Naturally, if the parameter is set to 0 the projector is off. Table 5.2 shows the relation between projector brightness and stops.

Brightness	0.25	0.50	1.00	1.80
Stops	-2	-1	0	+0.85
Lumen	100	200	400	720

Table 5.2: Project brightness and stops relation, Zivid One+ [86]

Zivid cameras have a thermal safety mechanism which will limit the maximum duty cycle of the camera, to ensure stable internal temperature. The safety mechanism can activate if the projector brightness value is >1.0 , and will pause the camera during capture.

Exposure time

The amount of time a image from a camera is exposed to light explains the term *Exposure time*, also known as shutter speed. Meaning how long the shutter remains open. In order to calculate depth, the Zivid camera captures multiple images. The minimum exposure time for the Zivid One+ camera is $6500\mu s$. When capturing a 3D image with Zivid camera's using the minimum exposure time, the camera use approximately $90ms$ for the capture. Image exposure of light will also be affected, when adjusting the exposure time. Needless to say, the acquisition time will be doubled if the exposure time is increased from e.g. $10\ 000\ \mu s$ to $20\ 000\ \mu s$. Also, the average image intensity will be twice as high.

Within a certain time window, Zivid is sampling information regarding light in the environment. Ambient light sources affects image acquisitions, and can be resolved. TV's and light source radiate structured light, which operates on similar sampling rates. Lamps connected to an AC power source in the EU, produces electrical voltage as a sine wave with a frequency of $50Hz$. Other countries, such as USA utilizes a frequency of $60Hz$. It is possible to observe flickering light when recording a video with $30fps$ in a garage or warehouse. The flickering light can be avoided if the capture rate matches the frequency of the ambient light source. Making the light appear constant. *Correlated sampling* [69] is the term used for the mentioned concept. It allows the user of 3D scanners to filter out noise which varies from an ambient light source. By choosing an exposure time t , dependent on frequency of a light source f and a positive integer n as seen in Equation 5.3.

$$t = \frac{n}{2f}, n = |Z| \quad (5.3)$$

Countries with $50Hz$ power frequencies must increase the exposure time value by $10000\mu s$ when adjusting the parameter [86]. Table 5.3 shows the recommended exposure times in relation with capture time and stops.

Exposure time [μs]	10 000	20 000	40 000	80 000	100 000
Capture time [ms]	130	250	490	970	1210
Stops	0	+1	+2	+3	+3.32

Table 5.3: Exposure time relation with capture time and stops

Gain

The parameter *gain*, also referred to as ISO in photography [86]. By adjusting the gain, it allows the user to configure the pre-amplification in the imaging sensor pixels. By increasing the gain, the images will appear brighter, because of the increase in sensitivity to light. If minimal time penalty is desired, increasing the dynamic range in a scene (for HDR acquisitions) with the gain would be beneficial. Very dark regions can be highlighted by adjusting the gain, however the gain amplifies everything the sensor sees and can be a drawback. It is recommended to keep the gain low, because higher gain reduces the peak signal to noise ratio. Even so, it is better to get noisy points, rather than no points at all. Table 5.4 shows the relation between gain and stops.

Gain	1x	2x	4x	8x	16x
Stops	0	+1	+2	+3	+4
db	-6	0	6	12	18

Table 5.4: Gain and stops relation [86]

5.3 Image capture

In this section, the acquisition settings was tested according to the knowledge acquired from Section 5.2. The captures were taken in a circular motion, with the camera pointing at the object and with the same distance for each capture. This motion was performed in order to achieve the same depth of focus for each capture. Images were saved with *OpenCV*, and implemented in the script for capturing [63]. *NOTE: "bgr8" message was used to get correct color format when saving the image.*

Nilsen [45] performed a comparison of three different RGBD camera, including Zivid One+ M. From his research, Nilsen acquired data for the Zivid camera's working range, which was 600 – 2000[mm]. Nilsen's data corresponds to Zivid's data [83]. For laser weld capturing, a mobile camera was utilized. Due to poor texture quality, and contrast deviation with the Zivid camera.

Initial conditions for image capture:

- The mean distance from the object to the camera was 800mm. Which was within the operating range for the Zivid camera.
- Power frequency of 50Hz, since this project was performed in Norway. Meaning the Exposure time acquisition setting, was adjusted according to subsection 5.2.
- The BM were always to be faced with the plastic cover (a) (Ref. Fig. 3.2) upwards. Therefore assuming it would be arriving the capture analysis in this position from a conveyer belt, or manipulator in an industrial practice.

Parameter tuning

Parameters were selected by the means of human- and histogram evaluation. A histogram is the distribution of colors in a 3D space, where the three dimensions is RGB [46]. Comparing color histograms is a typical method of comparing the color content of images. The method is based on the fact that images are typically represented as a series of pixel values, each of which corresponds to a different visible hue. For each image, color histograms are calculated to determine the relative proportions of pixels inside specified values. Similar images should have similar proportions of particular hues, according to the theory [16].

From Figure 5.1, the *f-number* should be approximately 5.6 – 6.0 since the mean distance from the camera to the object was 800mm. Settings were also evaluated with the lamps in the lab turned on, or off. For safety measurements, the parameter brightness was set to <1.0 for all tests. Due to the thermal safety mechanism. Six tuning adjustments were made.

Test 1

The chosen parameters settings for test number 1 can be seen in Table 5.5 From the first

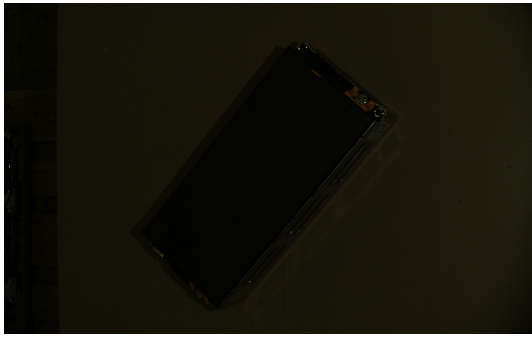
Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.8	1.0	10 000	1.0

Table 5.5: Acquisition settings, test 1

test the images were too dark, even with or without the lamps. However, to was possible to slightly see the shape of the BM.

Test 2

Test 2 had the same settings as utilized in Test 1, however the *Exposure time* was increased by 10 000 μs as seen in Table 5.6. From the second test the images appeared brighter compared to the first test as seen in Figure 5.5. It was also possible to see a deviation between the lamps turned on or off.



(a) With lamps turned on



(b) With lamps turned off

Figure 5.4: Test 1

Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.8	1.0	20 000	1.0

Table 5.6: Acquisition settings, test 2



(a) With lamps turned on



(b) With lamps turned off

Figure 5.5: Test 2

Test 3

From Test 2, the adjustment in *exposure time* worked well. Therefore, the same parameter was increased by 20 000 μs as seen in Table 5.7. With an *exposure time* of 40 000 μs , shape

Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.8	1.0	40 000	1.0

Table 5.7: Acquisition settings, test 3

and texture of the BM was significantly better as seen in Figure 5.6.



(a) With lamps turned on



(b) With lamps turned off

Figure 5.6: Test 3

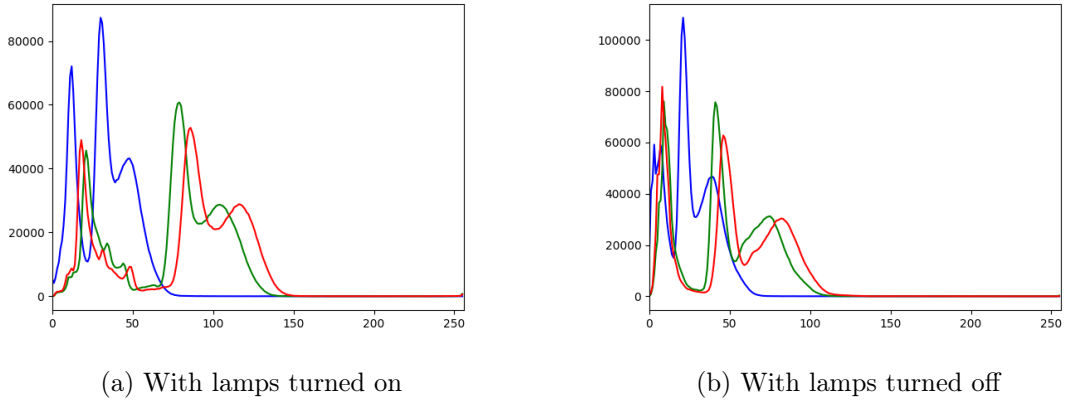


Figure 5.7: Test 3 histograms

Histograms were evaluated from this point, since the shape of the BM was relatively clear in the image. From the histograms in Figure 5.7, the settings with the lamps turned on had the RGB- values most evenly distributed along the RGB axis. Compared with the lamps turned off, the values was more attracted to the left.

Test 4

In test 4, the solution with the lamps turned on was discarded. Reducing the sources of reflectivity on the Battery Module. From test 3, the *gain* was adjusted in order to make the image appear brighter due to the projector's sensitivity of light.

Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.8	1.0	40 000	2.0

Table 5.8: Acquisition settings, test 4

Compared to test 3 with the lamps turned off in the lab, test 4 resulted in a brighter image as anticipated. From the histogram in Figure 5.8, it was also noticeable that the RGB-values was better distributed along the axis compared to test 3.

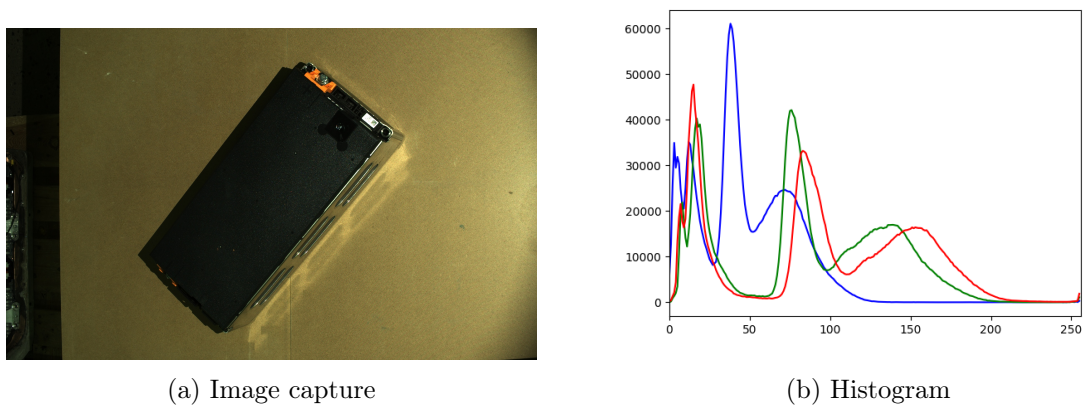


Figure 5.8: Test 4

Test 5

As seen in test 4, the shadows around the edges was not quite detailed with respect to the edge itself. In test 5, the *exposure time* was increased to $50000\mu s$, and the *brightness* was decreased to 0.5 seen in Table 5.9. The brightness was decreased due to reflectivity from the casing. From the image capture, the difference in reflection was noticeable. Also the

Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.8	0.5	50 000	2.0

Table 5.9: Acquisition settings, test 5

deviation between the edge and shadow was better. The histogram in test 5 is similar to test 4, however the R- and G-value had less distributed spikes. However, the B-value was moved slightly to the right.

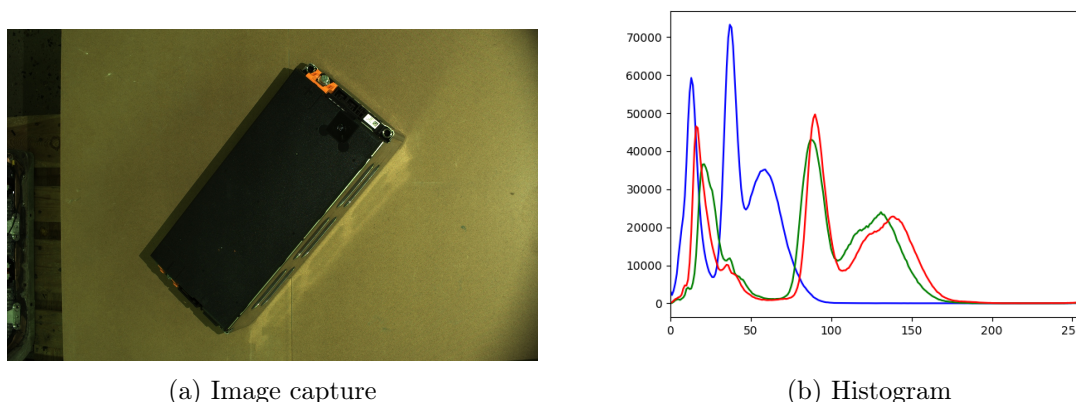


Figure 5.9: Test 5

Test 6

The last and final test with manual tuning. Compared to test 5, the *aperture* was adjusted to 5.6 and the *brightness* was decreased to 0.25. The aperture was adjusted in an attempt to enhance texture quality. From Figure 5.10, the image capture appear brighter compared

Parameter	Aperture	Brightness	Exposure time	Gain
Value	5.6	0.25	50 000	4.0

Table 5.10: Acquisition settings, test 6

to test 5. Also, the edges and shades deviated more. From the histogram, it was noticeable that the plot of the RGB-values had shifted more towards the middle.

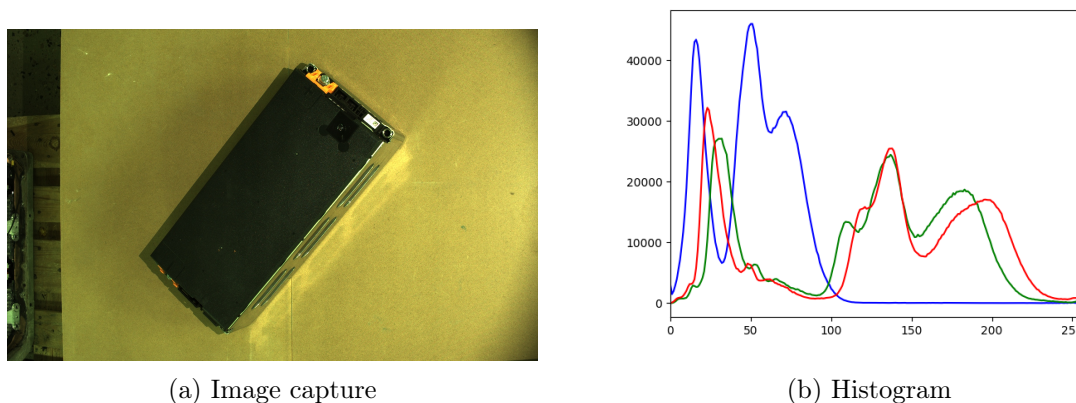


Figure 5.10: Test 6

Zivid studio

As mentioned in 5.1, Zivid studio provides an *analyse and capture* solution in their SDK. The software analyse the scene, and provide settings which can be utilized for capture. This

solution was tested, for comparison with the manual tuning. The analyzed solution overlays 4 acquisitions settings in order to provide the best capture. The settings were exported as a .yaml file, which then was imported in the *load_settings_from_file* service.

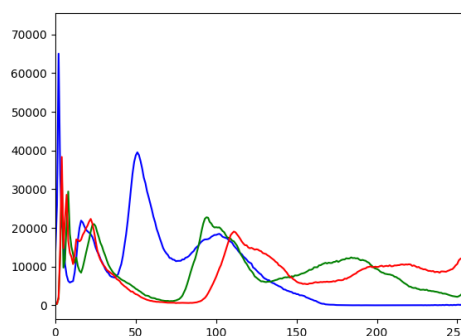
Parameter	Aperture	Brightness	Exposure time	Gain
Acquisition 1 values	5.26	1.8	6500	1.05
Acquisition 2 values	3.59	1.8	6500	1.45
Acquisition 3 values	2.86	1.8	6500	2.73
Acquisition 4 values	2.58	1.8	13 000	3.75

Table 5.11: Acquisition settings, Zivid Studio

From the image seen in Figure 5.11 the overall texture quality was more realistic compared to test 6. The histogram had its RGB-value even more distributed along the axis as well, however the far left R- and G-value explains the yellow perception of the image.



(a) Image capture



(b) Histogram

Figure 5.11: Zivid SDK analyzed capture

Evaluation

From these tests, it would be reasonable to continue with the settings from the Zivid SDK. However, images from test 6 happened to have better deviation between shades and edges. The deviation compared with the Zivid SDK and manual acquisition settings tuning can be seen in Figure 5.12.



(a) Zivid SDK



(b) Manual tuning

Figure 5.12: Comparison with Zivid SDK and manual tuning

For further evaluation, the plastic cover (a) and the shadow were compared in contrast with a simple script utilizing *OpenCV*. Some contrast evaluation methods are Weber contrast, Michelson contrast and Root-Mean-Square (RMS) contrast. The method utilized for evaluation in this project was the RMS contrast. The angular frequency content or the spatial

distribution of contrast in the image have no effect on RMS contrast. The standard deviation of pixel intensities is defined as RMS contrast [54]. RMS contrast can be defined as in Equation 5.4. x_i lays in the interval $0 \leq x_i \leq 1$ and is the level of gray, normalized. The mean normalized gray level, \bar{x} can be defined as seen in Equation 5.5.

$$RMS = \left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{\frac{1}{2}} \quad (5.4)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.5)$$

By cropping the images, visualisation of the shade and plastic cover could be further processed. The images were converted from RGB images to greyscale with the function below:

```
grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

The standard deviation of the pixel intensities in the greyscale images were then calculated:

```
contrast = grey.std()
```

From the test, grey-scale contrast values were gained. For reference, a horizontal splitted image with white on top and black at the bottom was utilized. The results can be seen in Table 5.12. If the value is zero, there would be no deviation in the contrast.

Zivid SDK Cropped	Manual tuning cropped	Black and white image
6.52	11.70	123.39

Table 5.12: Grey-scale contrast values

For the purpose of labelling in regards of object detection and training, the manual settings from test 6 would be the best candidate considering the higher contrast of the plastic cover (a) and the shadow.

Chapter 6

Object detection

Predicting the presence of one or more object is the term used for object detection within computer vision. This chapter presents the chosen object detection model Yolov5, additional software, dataset generation as well as training procedure for the custom model.

6.1 YOLO

YOLO, or *You Only Look Once* is a state-of-the-art object detection algorithm which divides images into a grid system. Due to its speed and accuracy, YOLO is considered one of the most famous object detection algorithms [73]. As seen in Figure 6.1, YOLOv3 and YOLOv4 outperforms other object detectors in the terms of speed [FPS]. Several versions of YOLO

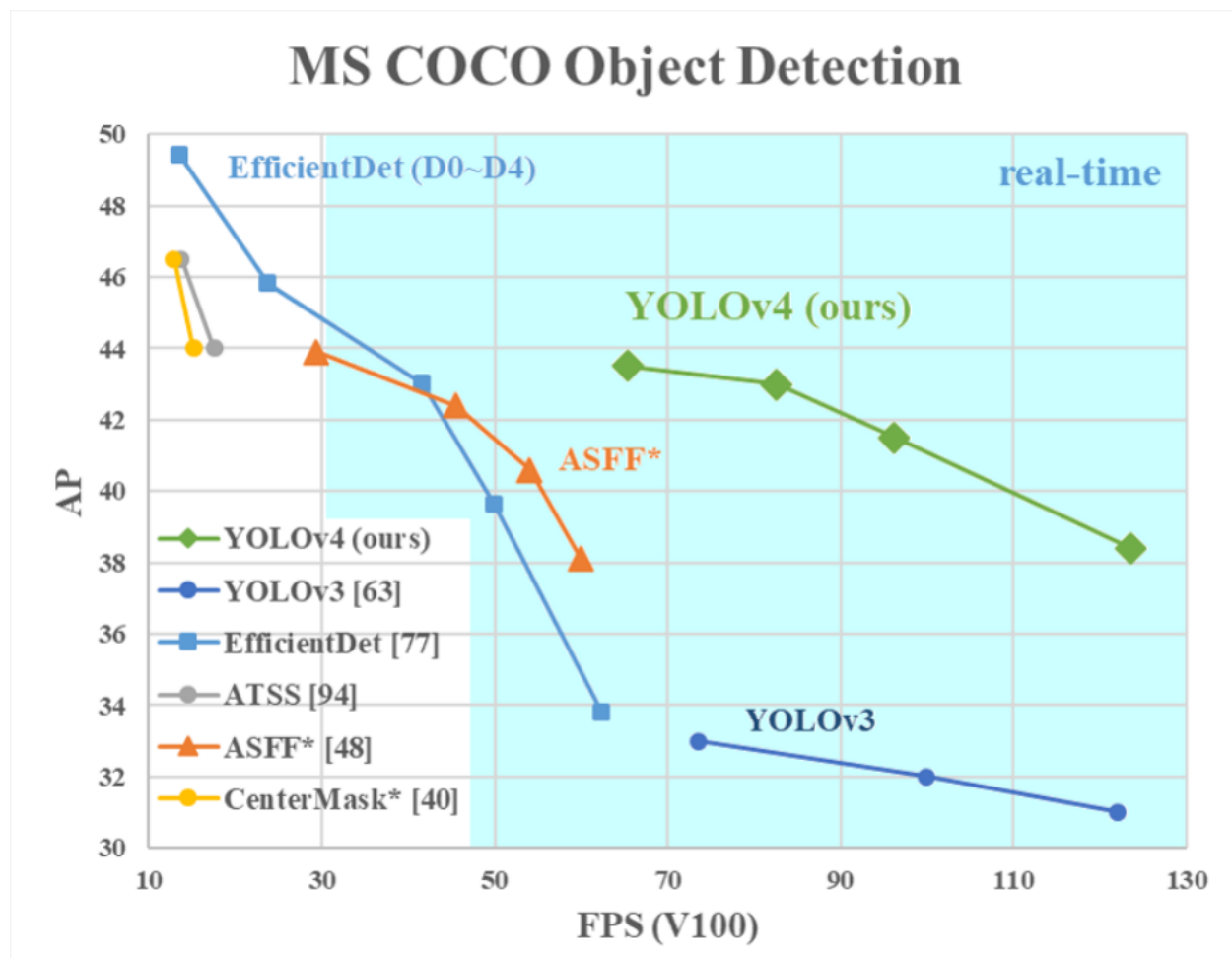


Figure 6.1: Comparison of Yolov4 and other models in terms of frame rate and accuracy [42]

have been developed, some of them are listed in Table 6.1. Including the versions listed in

Version	Author	Release date
v1	Joseph Redmon	8 Jun 2015
v2	Joseph Redmon and Ali Farhadi	25 Dec 2016
v3	Joseph Redmon and Ali Farhadi	8 Apr 2018
v4	Alexey Bochkovskiy, et. al.	23 April 2020
v5	Glenn Jocher	18 May 2020

Table 6.1: YOLO History [73]

Table 6.1, there have been released some newer versions of YOLO. PP-YOLO and Scaled Yolov4 were released late 2020, and PP-YOLOv2 was released in April 2021 [43].

Comparing YOLO v5 with all other version, version 5 is utilizing a PyTorch implementation rather than Darknet which is primarily used on the other versions. Darknet is written in C and CUDA, and is an open source neural network framework. YOLOv3 has been developed with both PyTorch implementation and Darknet. Yolov5 is said to be the fastest amongst the object detectors. As seen in Figure 6.2, YOLOv5 is considered faster in terms of GPU computing and average precision compared to EfficientDet. From Figure 6.1, EfficientDet appeared to be more precise than YOLOv4. V5 is also lightweight compared to v4. A weights

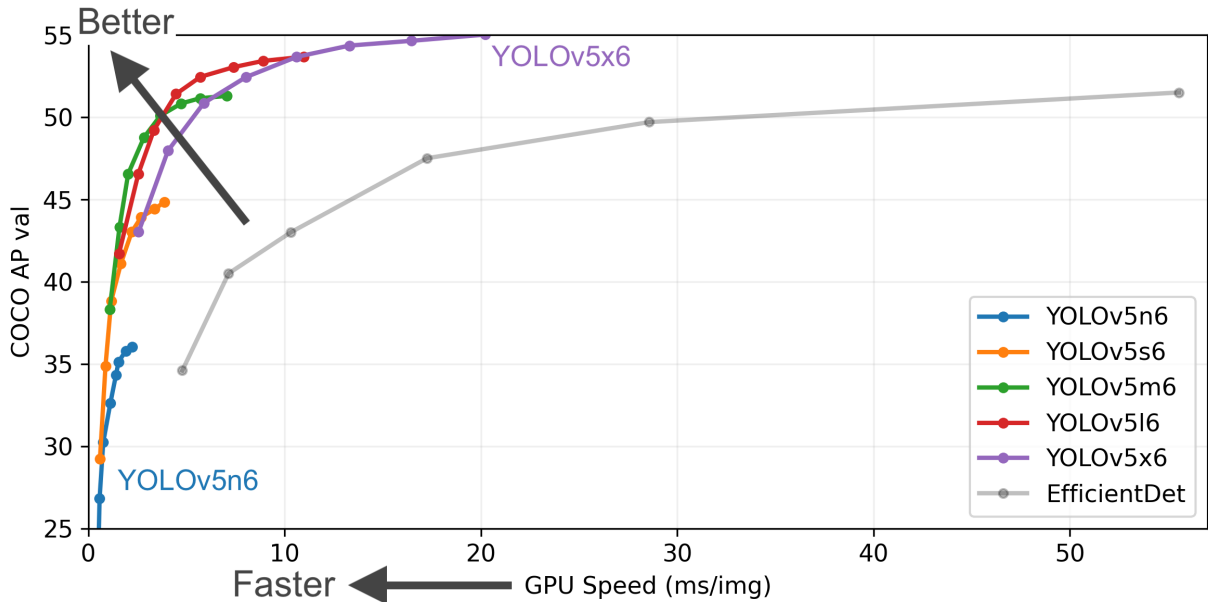


Figure 6.2: Comparison of Yolov5 and EfficientDet in terms of GPU speed and average precision (AP) [42]

file from YOLOv5 was 27 MB, while a YOLOv4 weights file with the darknet structure was 244MB. YOLOv5 is therefore 90% smaller than YOLOv4 [68]. Five models was released with YOLOv5 as seen in 6.3. The larger models like YOLOv5l and -v5x will produce better results in most cases, but require more CUDA memory and will take longer to train. YOLOv5n and -v5s are recommended for mobile applications. YOLOv5s is utilized by default [74].

As mentioned, YOLOv5 utilize PyTorch implementation. PyTorch is a framework for deep learning which is open-source, and known for its flexibility and easy-to-use. Also, due to its compatibility with Python programming language, it is favored by machine learning developers and data scientists. PyTorch is used for building deep learning models, which is a type of machine learning. Machine learning is common in applications such as image recognition and language processing [47]. PyTorch is GPU compatible and utilize reverse-mode-auto-differentiation [65], enabling computation graphs to be modified on impulse.

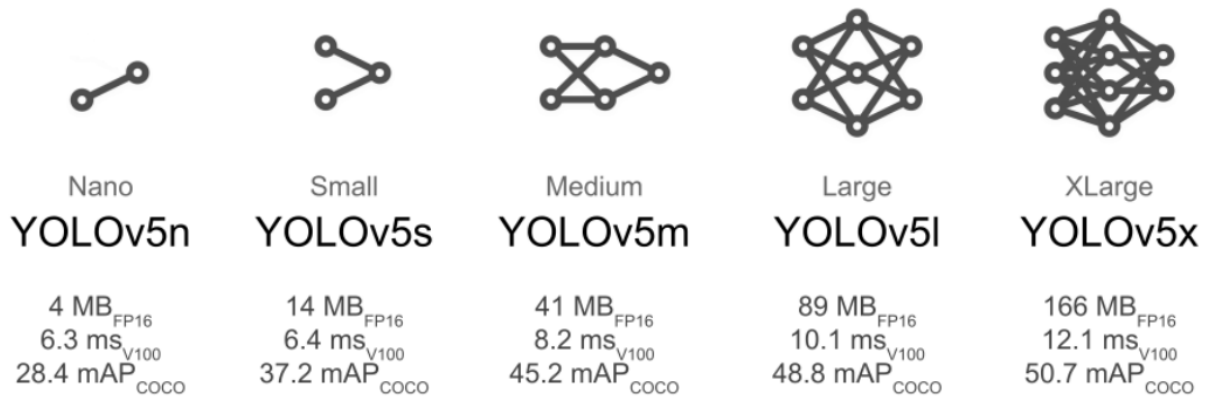


Figure 6.3: Released models of YOLOv5 [74]

From Ultralytics Github [72], all necessary dependencies listed in *requirements.txt* must be installed in order to work with YOLOv5. In addition to the dependencies, PyTorch, Python 3 and CUDA/cuDNN must be preinstalled. CUDA was created by Nvidia, and is a parallel computing platform and programming model. CUDA harness the power of GPU accelerators and helps users to speed up their applications [51]. Nvidia CUDA Deep Neural Network library (cuDNN) provides libraries which is GPU-accelerated to be utilized for deep neural networks. Operations such as convolution, pooling and activation layers can be solved using cuDNN's highly tuned implementations. Deep learning frameworks such as Caffe2, Chainer, Keras, MATLAB, MxNet, PaddlePaddle, PyTorch and Tensorflow are supported by cuDNN, and accelerates computing compared to low-level GPU performance tuning [21]. Version 11.4 of CUDA is the proper decision when comparing version compatibility with GPU driver [49]. With CUDA version 11.4, the corresponding cuDNN version which should be installed is v8.4.0. In comparison to the CUDA version, PyTorch version 1.10.2 is the natural choice. Version 3.7.0 \leq of Python is listed as the requirement YoloV5, however Python Version 3.6.9 worked in this scenario. See Appendix E for workflow with software versions.

6.1.1 Annotation & dataset generation

In order for the object detector to recognize the desired objects, it must be supervised with bounding box annotations. There are labeling tools online, e.g. LabelImg, CVAT, and VoTT [66]. The online annotater utilized for this project was Roboflow.ai. Roboflow a simple web-based tool for labeling and managing you images then to be exported in YoloV5's annotation format. It is important to mention that Roboflow's export service is only available when a project is "public". Two datasets were generated:

- **Dataset 1:** Battery module (BM) and laser weld
- **Dataset 2:** Ref. Fig. 3.2, the sleeve where the screw is connecting the modules (-)

The laser weld, and the "sleeve" were chosen with respect to dismantling concepts for the compressive plates (c), ref. Fig. 3.2.

Yolov5's labeling format is supported by many of the annotation platforms. The format provides one *.txt* file per image, and contains data for each bounding box for an object. Bounding box annotations are very common. The value of the annotations are within the range of 0 to 1, and are normalized to the image size. Annotations are generally represented as shown below:

```
<Object class> <X center> <Y center> <Box width> <Box height>
```

```
0 0.383 0.439 0.183 0.628
0 0.507 0.454 0.191 0.713
```

Figure 6.4: Example .txt file for two objects in one image [8]

If there are two objects to be detected in one image, the .txt file might look like shown in Figure 6.4. Roboflow provides the option to annotate images with *Polygon/segments* instead of the bounding box. Polygon annotations do not have the same format compared to the bounding box format. Rather than x,y center and box width/height, it provides x- and y coordinates of the vertices in addition to the corresponding object class. Two labels were made, BM (Battery module) and weld. These were both annotated with the polygon option. Images utilized for a custom model should be resized for computational efficiency. Smaller image file sizes, will result in faster training.

Augmentation or data augmentation is the term used for a strategy used in object detection. Practitioners utilize augmentation in order to increase the diversity of the data available for the training model, without collecting new data. Augmentation techniques for **Dataset 1** utilized in this project *Image flip*, *90 °rotation*, *General rotation (15 °)* and *Shear (15 °horizontal and 33 °vertical)*. The same augmentation techniques listed above were utilized for **Dataset 2** as well, in addition to *Saturation*, *Brightness*, and *Exposure*.

Roboflow exports .txt annotations and .yaml configurations in the desired format, in this case the YOLOv5 PyTorch format as seen in Figure 6.5. An option could also be to manually prepare a dataset, but this would naturally be more time consuming.

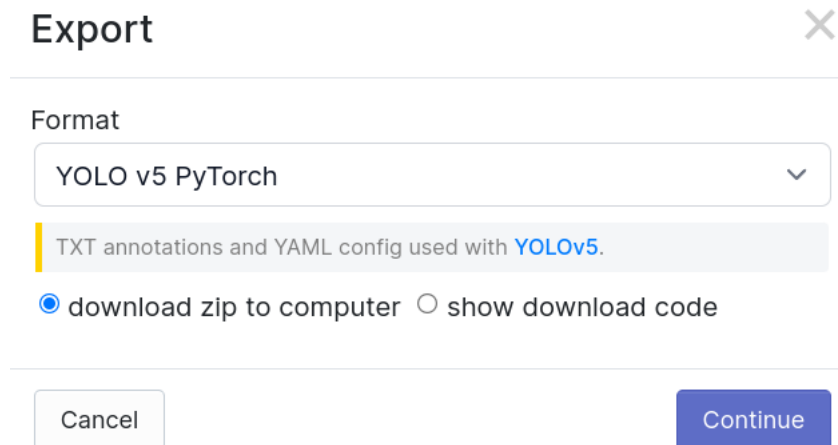


Figure 6.5: YOLOv5 Pytorch export

Data directories to be utilized for YOLOv5 are provided in the following structure as seen in Figure 6.6.

6.1.2 Training

When training the YOLOv5 algorithm to recognize your desired objects, there are some parameters which can be specified when launching *train.py*. These parameters are:

- Epochs
- Image size
- Batch size



Figure 6.6: Data directory structure [8]

- Hyperparameters

Epochs

Epochs is the term used for the number of times the machine learning algorithm will go through the entire dataset. Each sample in the training dataset updates the internal model parameters, this is equal to one epoch. An Epoch could include data of one or more batches [14]. When training your custom model, it is recommended to start with 300 epochs with Yolov5 [74]. The number of epochs should be reduced, if the model overfits at an early stage. Train for more epochs, i.e. 600, 1200 etc. if overfitting doesn't occur after 300 epochs. The training data for the model can be too well, this is the term *overfitting*. This Overfitting occurs when the detail and noise in the training data is trained for to many iterations, and impacts the performance on new data negatively. In other words, noise and random fluctuations from the training data are learned as concepts by the model. These concepts does not apply to new data, which will affect the model to generalize. In addition to overfitting, there is underfitting and a good fit. The term *underfitting* can neither model the training data, or new data. An underfitted model will have poor performance on the training data, and is not suitable. The term *good fit* is the sweet spot between overfitted and underfitted models. In practice, this can be very difficult to achieve [13]. Techniques which can be used to limit overfitting:

- Resampling techniques to estimate accuracy of the model
- Restrain a validation set.

Image size

By default, the Common Objects in Context (COCO) dataset is training at a resolution of `-img 640`. However, there is a high amount of small objects in the COCO dataset and it could be beneficial to train COCO at a higher resolution, e.g. `-img 1280`. When training a custom model, the default resolution would suffice the training. However, if there are many small objects, higher resolution could be preferred. It is also beneficial to train your model at the same resolution as the detecting resolution [74].

Batch size

The number of samples passing through the dataset before updating the internal model parameters is known as *Batch size*. From the expected output variables the predictions are put to comparison after a batch, and an error is calculated. The model is improved from the update algorithm calculated from the mentioned error. Meaning the error gradient decreases. Batch gradient descent is a term used when the all the training samples are used to create one batch. If the batch size equals to one, the learning algorithm is called Stochastic gradient descent. Mini-batch gradient descent is when the batch size is greater than one, but smaller than the size of the training set [14].

- **Batch Gradient Descent:** Batch size = Size of training set

- **Stochastic Gradient Descent:** Batch size = 1
- **Mini-Batch Gradient Descent:** $1 < \text{Batch Size} < \text{Size of Training Set}$

It is recommended to use the largest *-batch-size* which the hardware allows. Poor batchnorm statistics is the result of small batch sizes and should be avoided [74].

Hyperparameters

Ultralytics recommends to train your custom model with the default hyperparameters [74]. Overfitting will be reduced and delayed by increasing augmentation hyperparameters, allowing for longer training and higher final mean-average-precision (mAP).

Training of custom model

When training your custom model for object detection, data has to be distributed between *train*, *validation* and *test*. Roboflow recommends a 70-20-10[%] split [67]. The train, validation and test split utilized for this project was 79-17-5[%] for Dataset 1, and 67-22-10[%] for Dataset 2. A total of 145 images were distributed between train, validation and testing in Dataset 1, and a total of 58 images for Dataset 2. YOLOv5's GitHub page [72] provided a procedure on how to train a custom model, and the dataset was tested with different training parameter values of the ones mentioned above. The hardware specifications on the laptop utilized in this project can be seen in Appendix E. The hardware of your computational unit affects the training in terms of speed, the better hardware the faster the algorithm will train your model. Specially the GPU, since CUDA/cuDNN make use of GPU accelerators.

When training a custom model, YOLOv5 displays a message when the results does not improve after 100 *Epochs* as seen in Figure 6.7. Adjusting the number of epochs when this warning appear, it is less likely that overfitting will occur. The default resolution value of

```

Epoch      gpu_mem      box      obj      cls      labels  img_size
337/599    4.15G    0.02813  0.006949  0.0004519    38      640:
          Class      Images      Labels      P      R      mAP
          all        23         23      0.911    0.882    0.911
Stopping training early as no improvement observed in last 100 epochs.
To update EarlyStopping(patience=100) pass a new patience value, i.e.

```

Figure 6.7: No further improvements in the last 100 epochs

image-size is 640x640. Down-scaling the images equals smaller data per image, resulting in faster training. It is important to mention that the detection should be performed in the same resolution as the training. Maximum batch-size which could be utilized for training with the hardware was 32. Execution command for training when directory is changed to yolov5 folder can be seen below:

```

~$ python3 train.py --epochs 200 --batch-size 32 --data libres2.yaml

```

Training parameters utilized, and the duration of the training when the number of epochs were completed for the custom model with generated dataset 1 in this project can be seen in Table 6.2. Training results from the custom model utilized in this project will be further evaluated in Chapter 7.

Metrics and losses from training

Fig. 6.8 displays an example of the results from a custom trained model.

Epochs	Batch-size	Image-size	Hyperparameters	Time [hours]	No. of images
200	32	640	hyp.scratch-low	0.1	114

Table 6.2: Training parameters

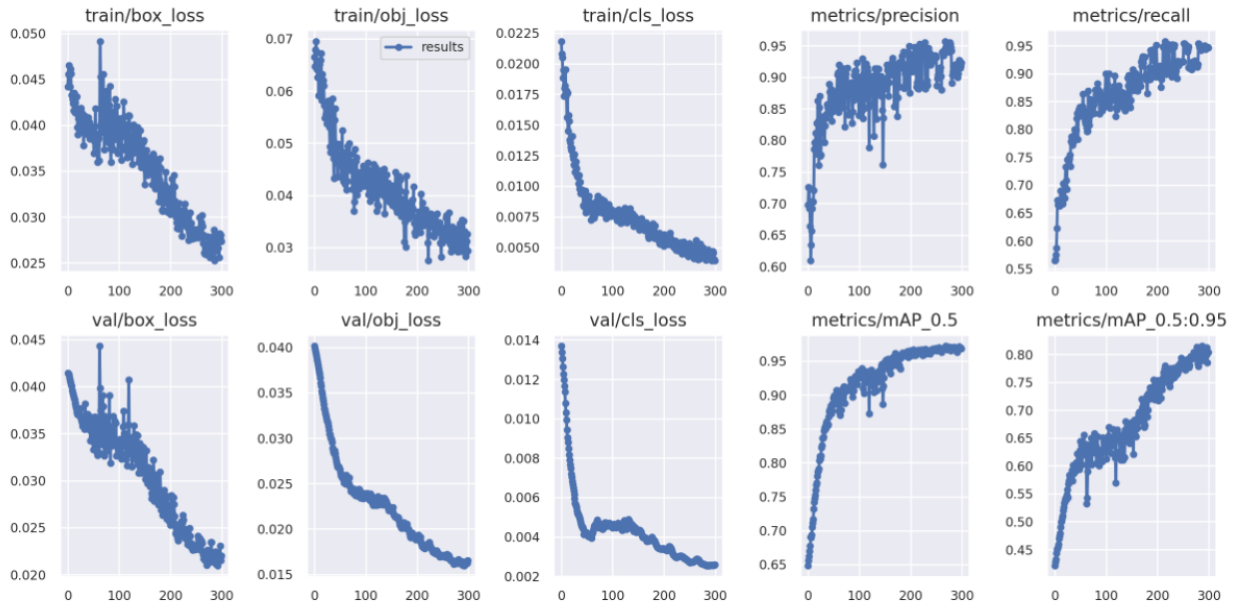


Figure 6.8: Example of training results from YOLOv5 [75]

In order to understand the results better, it is necessary to summarize the metrics and losses from the training. A custom trained model can predict classes which are either positive or negative, and the predictions may be correct/true or incorrect/false. As for an example, if a custom object detection model were to detect "trees" or "no trees", a positive class would be "trees" and "no trees" would be negative. If the prediction is correct, the prediction is considered *true*, whilst an incorrect prediction would be *false*. From Figure 6.9 a positive prediction corresponds to the red bounding boxes, meaning the algorithm predicted a tree in the image. A negative prediction corresponds to the dark blue bounding boxes, where the algorithm predicted that there is no tree in the image.

In Table 6.3, each highlighted number from Figure 6.9 is clarified.

Number	Description
1	True positive: The algorithm correctly predicted the presence of a tree.
2	False positive: The algorithm incorrectly anticipated the presence of a tree.
3	False negative: The algorithm incorrectly predicted that there is no tree.
4	True negative: The algorithm correctly predicted that there is no tree.

Table 6.3: Explained numbers from Figure 6.9

The quality and number of training samples, the imagery input, parameters of the model, and the accuracy requirement threshold all influence the accuracy of an custom object detection model. To determine if the predictive outcome is either true positive or false positive, an calculation of the Intersection over Union (IoU) ratio is performed. IoU is a metric that measures the amount of overlap between the bounding box around a predicted object and the bounding box around ground reference data [55]. Referring to Figure 6.10. **1** is equal to the anticipated bounding box and the ground reference bounding box intersection in this area. **2** is the combined surface area of the anticipated bounding box and the ground



Figure 6.9: Positive- and negative tree predictions [55]

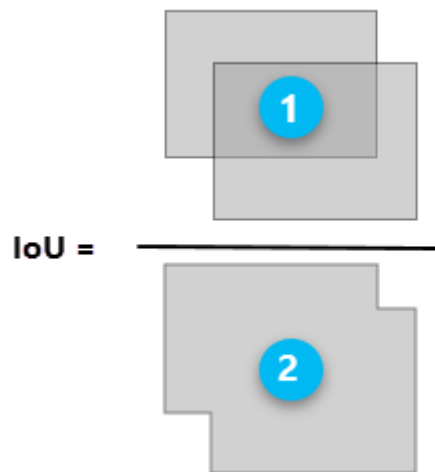


Figure 6.10: IoU distribution [55]

reference bounding box. Therefore IoU can be formulated as seen in Equation 6.1.

$$IoU = \frac{Intersected\ Area}{Union\ Area} \quad (6.1)$$

There are primarily three kind of metrics provided from the YOLOv5 training result, *Precision*, *Recall* and *mean-Average-Precision (mAP)*. The ratio from the number of true positives, and the number of positive predictions is the described concept for precision. E.g. if the precision equals 90 percent, the total detected objects could be 100, where 90 of them were correct. Precision can be formulated as seen in Equation 6.2.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6.2)$$

The quantity of true positives divided by the total number of genuine (relevant) objects is known as recall [55]. E.g if the recall is 75 percent, it could correspond to 75 correctly

detected objects when there is actually 100 objects in the image. The formula for Recall can be seen in Equation 6.3.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (6.3)$$

The last metric is the mAP. The average AP over various IoU thresholds is the concept of mAP. E.g. from the YOLOv5 training results, the mean Average Precision (mAP) at the IoU criteria of 0.5 is called mAP 0.5. The average mAP over several IoU thresholds, ranging from 0.5 to 0.95, is mAP 0.5:0.95 [8].

There are three different types of losses [34]:

- **Box loss:** Represents how effectively the algorithms bounding box covers an object, and how well it can predict the centre of an object.
- **Objectness loss:** Objectness is a probability measure for the presence of an object in a suggested zone of interest. If the objectivity is high, the image window is more likely to include an object.
- **Classification loss:** The classification loss indicates how well the algorithm can predict the exact object class.

Validation

For your custom trained model, Ultralytics provides a validation script. Parameters like batch size and image size can be set just like the training script. Validation should be performed with the same batch size and image size as training. Weights from the custom trained model should be implemented. The command to start validation with specific parameters from the terminal is shown below. These are the parameters that were used in this project.

```
~$ python3 val.py --batch-size 32 --img-size 640 --data libres2.yaml ...  
--weights runs/train/exp11/weights/best.pt
```

Running the validation script outputs a precision-recall curve. An example can be seen in the example in Fig. 6.11. A plot of precision (y-axis) and recall (x-axis) may be used to assess an object detection model's performance. If the precision remains high as the recall grows, the model is called a good predictive model [55].

Detection

By executing the *detect.py* as seen below, the script detects object from the custom trained model from Dataset 1. *-source* could be specified as a folder where image captures are saved, or webcam if it is desired. Multiple command specifiers can be seen in the *detect.py* script from Ultralytics Github [72].

```
~$ python3 detect.py --data libres2.yaml --img-size 640 --weights ...  
/home/josteink/MAS500G1/yolov5/runs/train/exp11/weights/best.pt ...  
--source /home/josteink/MAS500G1/yolov5/data/images
```

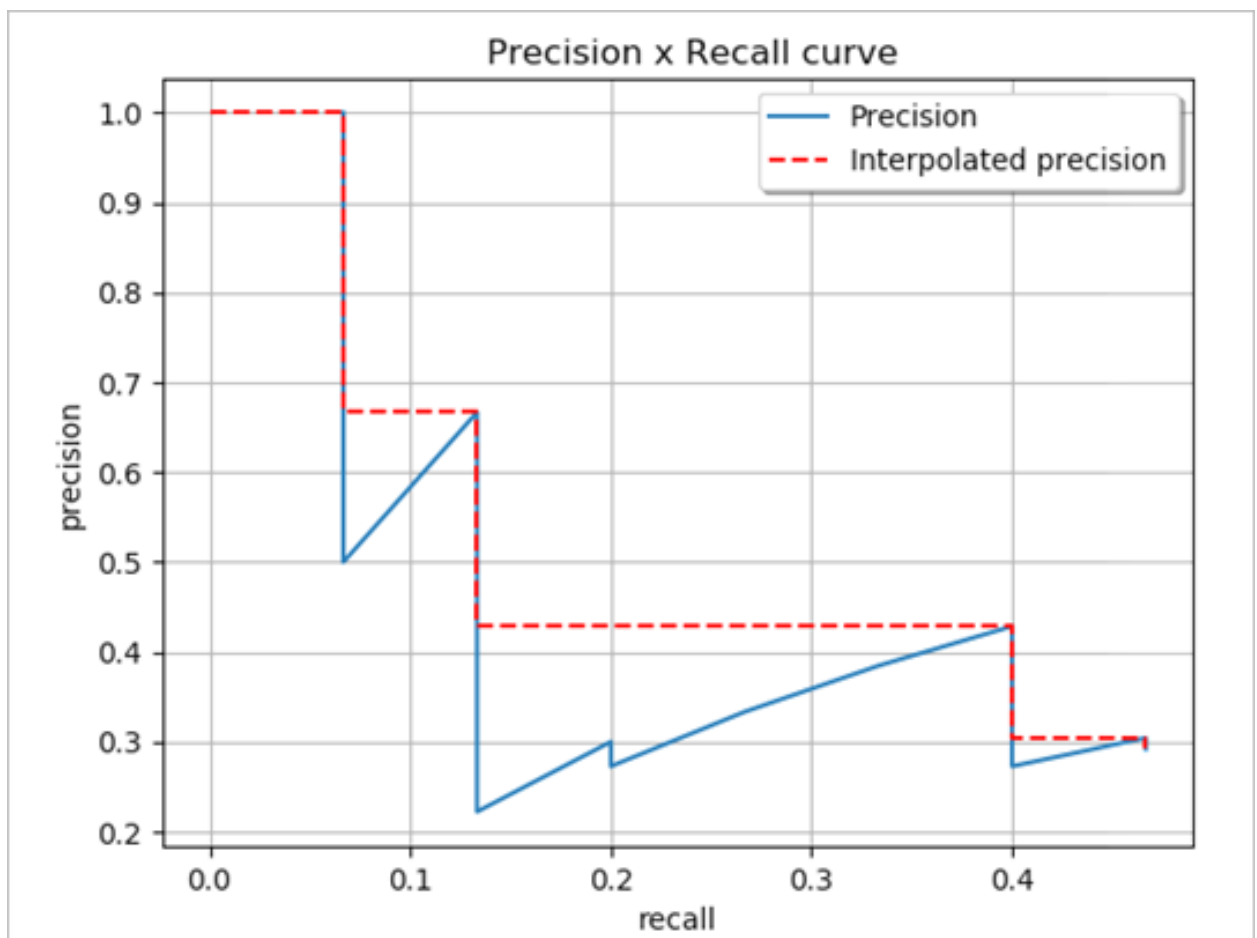


Figure 6.11: Example of an precision-recall curve [55]

Chapter 7

Project Results

The findings of the FEM-analysis of the battery module are presented in this chapter, along with the results of the object detection algorithm's training. The results of the Fem-analysis of the pins are presented in chapter 3.

FEM-analysis results

The results for all four configurations are represented in Fig.7.4 showing the Von Mises stress and deformation while Fig.7.5 is showing the displacement. Table 7.1 lists the maximum values of stress and displacement. The von Mises yield criterion states that if the stress in a material when loaded is equal to or greater than the yield strength of the material then the material will yield [31]. The configuration of long pins on one side resulted in the highest value of stress (5042 MPa), as shown in Fig. 7.1. Even if the highest value of stress accumulated at a sharp edge which could indicate inaccurate prediction, Fig 7.1 shows stress accumulation above 800 MPa beside the weld throughout the entire height.

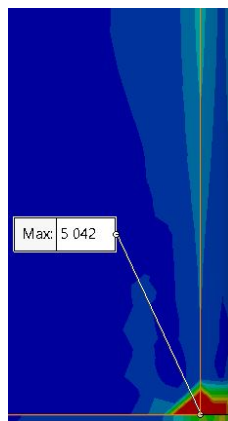


Figure 7.1: Maximum stress location

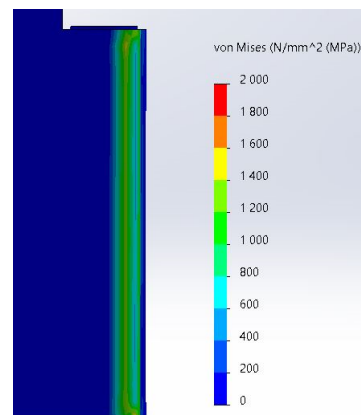


Figure 7.2: Stress accumulation on the weld

The values of the von Mises stress are exceeding the yield strength of the material (172.37 MPa) by a huge margin which indicates that the material will not only endure elongation but will eventually fracture as seen in Fig.7.3. As mention earlier, the weld is the weakest point and the analysis along with previous research indicates that the weld will fracture when the load is applied.

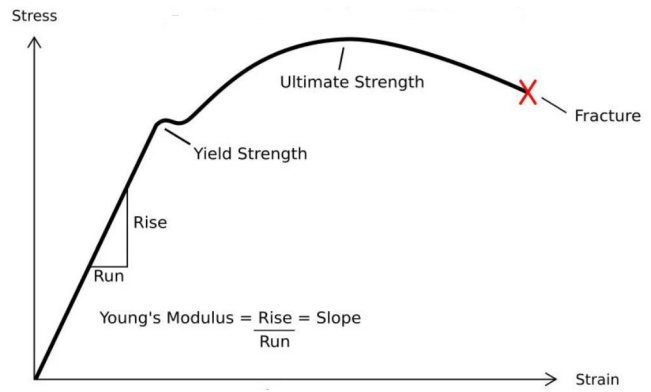


Figure 7.3: Stress-strain curve

Configuration	Von Mises stress [$\frac{N}{mm^2}$]	Displacement [mm]
Long pins	3959	3
Short pins	3141	2
Long pins one side	5042	2
Short pins one side	3211	1

Table 7.1: FEM-analysis results

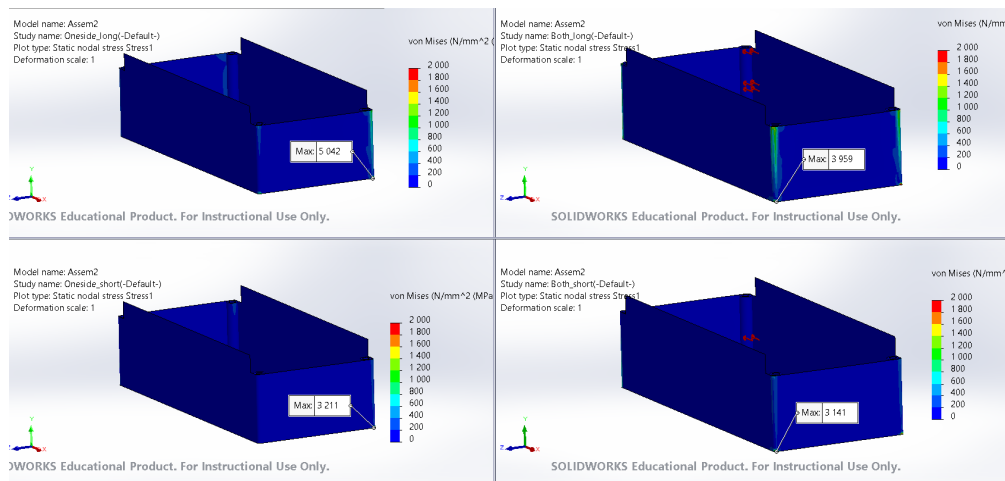


Figure 7.4: Von Mises Stress

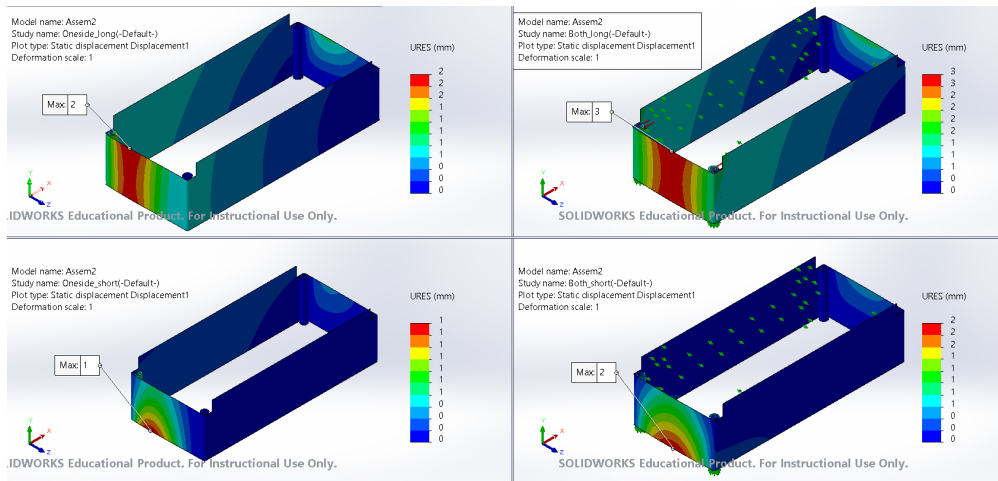


Figure 7.5: Displacement

Training results

The metrics and losses from Dataset 1 can be seen in Fig. 7.6, and the Precision-recall curve in Fig. 7.7. Overall the results are representative. From the losses, it is observable that the algorithm improves its bounding box-, objectness- and classification (cls) loss. The metrics however show some oscillations at the early stage of training, but became more stable after 150 epochs.

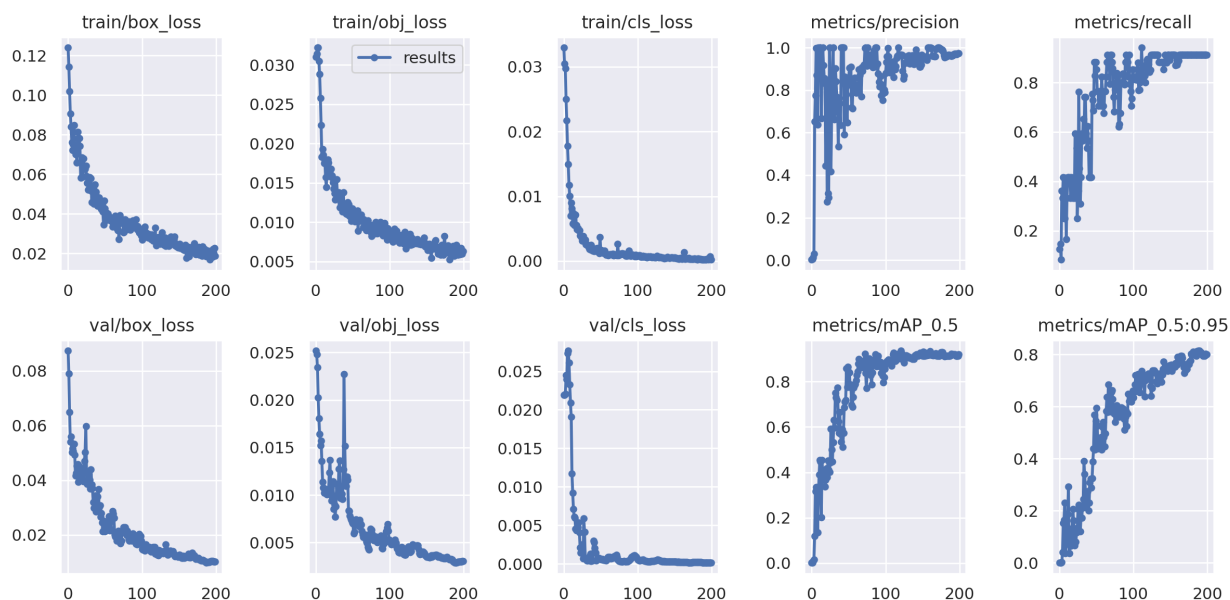


Figure 7.6: Metrics and losses from Dataset 1

The precision-recall curve indicates an acceptable model for all classes since the precision decreases after the recall pass 0.8. It is important to mention that the curve drops with respect to the class "Weld". Precision of the class "BM" on the other hand remains high, while the recall increases. The dark blue curve is combined from the class "BM" and the class "Weld".

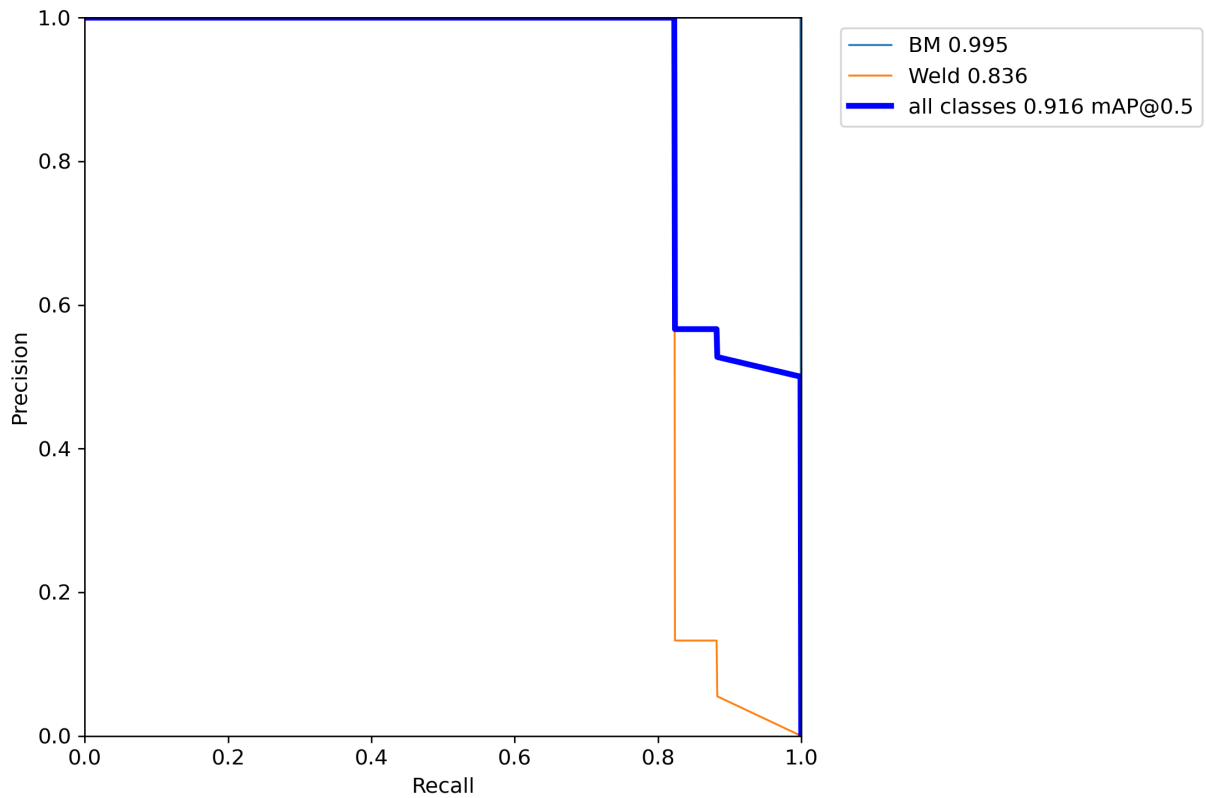


Figure 7.7: Precision - recall curve, Dataset 1

Dataset 2 however, did not provide just as good training results compared to Dataset 1. From Fig. 7.8 it is possible to identify that the algorithm improves but does not reach a defensible state. Argumentation on how this model could be further improved is discussed in Chapter 8

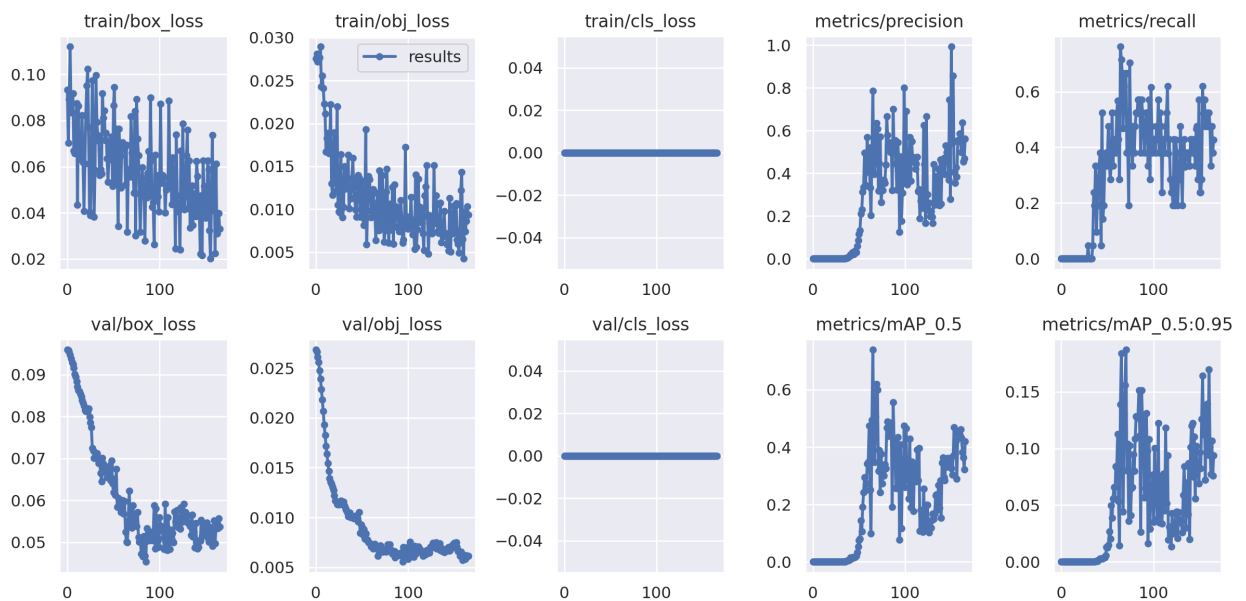


Figure 7.8: Metrics and losses from Dataset 2

Component detection

The two generated datasets were tested for object detection. To achieve some form of generalization between the images from training and the detection, captures for detection were captured from angles which was not utilized for training. Different configurations with lights turned on and off, and acquisition settings were also tested. From Dataset 1, the laser welds were not able to be detected.

From Fig. 7.9, the Battery Module achieved a score of 0.96. The detection was performed at an 25° angle deviating from the training captures.



Figure 7.9: Detection of BM with tuned settings

In order to test the generalization of the algorithm, the BM was oriented with the plastic cover (a) to the right. Also with the light in the lab turned on. As seen in Fig. 7.10, the detection achieved a score over 90% and the bounding box covers the whole BM. From Fig. 7.11, the acquisition settings from the "analyze and capture" service in Zivid studio were utilized, the lights were turned off. As discussed in 5.3, the contrast deviation between the plastic cover (a) and the shadow is too small. From the bounding box, it is possible to notice that it also covers the shadow.

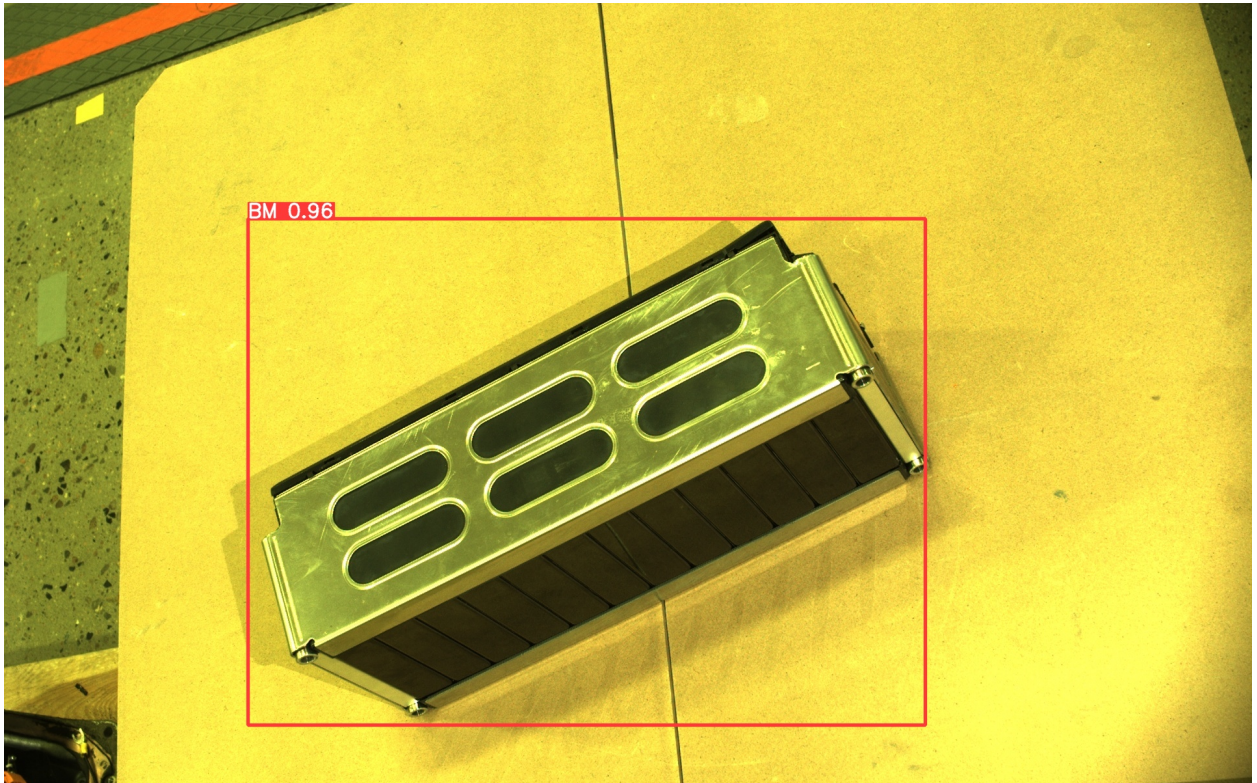


Figure 7.10: Plastic cover (a) not facing upwards with lights on, tuned settings

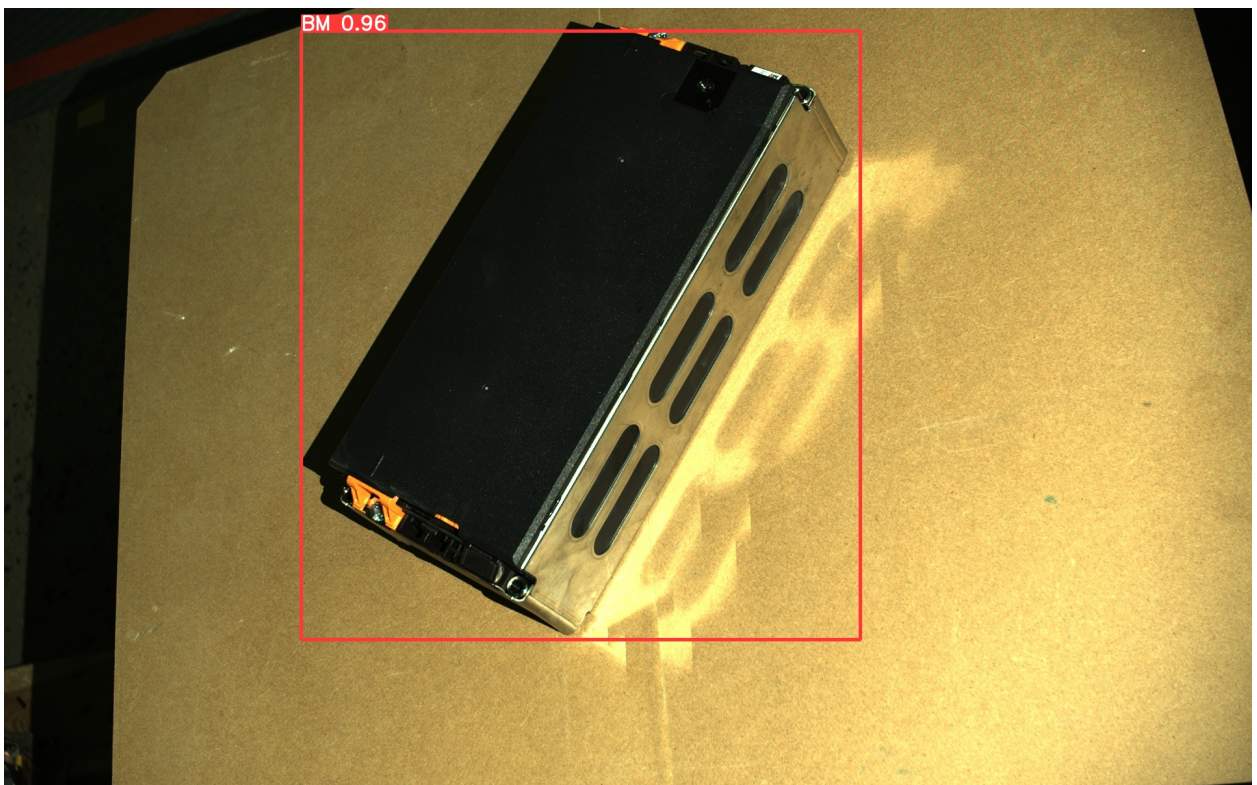


Figure 7.11: Zivid SDK settings

The Sleeve can be observed from Fig. 7.12 to be detectable. However, the score achieved can be observed to be at a precision of 0.33%. Dataset 2 was generated by a short submission deadline, and will be further discussed in Chapter 8.



Figure 7.12: Sleeve detection from Dataset 2

Chapter 8

Discussions

Hydraulic unit

Taking the pins into consideration the highest stress producing configuration might not be the best option, since all configurations exceeded the yield strength of the material with margin of at least 2969 MPa. Then it would be convenient to use the configuration that produce enough stress to fracture the welded compressive plates, but at the same time is low enough to ensure repetitive use of the pins without permanent deformation.

Further study is needed to predict the failure of the compressive plate structure, using material testing and numerical simulation as for example ANSYS or Abaqus/Explicit. The material behavior for strains greater than the ultimate tensile strength is complex due to geometrical and material nonlinearities, stress triaxiality, and strain localization followed by material failure. Many aspects of numerical modeling are still under development to be able to apply it on practical applications [36]. However, to facilitate fracture of the welded compressive plates, one could consider to initiate a localised pit or crack that will propagate when applying tension as this is the case with pitting corrosion. This initial pit could be generated by the mean of special tool attached to a robotic manipulator.

Milling

The milling option is fast, reliable and cost-effective. As mentioned in sec. 3.4.2 the available robot (IRB4400) is not designed for milling and a stronger type must be considered. The tool wear would be negligible compared to the pins in the hydraulic unit concept. However, we must obtain precise position- and orientation data. This is done by implementing pose estimation technique in the computer vision algorithm.

Object detection

From Fig. 6.3, there are 5 different models of Yolov5 which can be utilized. In this project the default model was utilized for training, Yolov5s (v5s). v5s is a rather small model, but has its advantages as well. The smaller model, thus shorter time for training. If a larger model (e.g. Yolov5x) was chosen, the mean-average-precision (mAP) may be improved by up to 13%.

Data set 1 included a total of 145 images, while Ultralytics recommends a minimum of 1500 images per class [74]. The BM were even tested with different configurations in order to test the generalization of the model. The results deviated. It was observed that the algorithm detected the shadows from the BM as the BM itself with the Zivid SDK settings. From the contrast deviation explained in 5.3, one should train the algorithm for the usage purpose.

Meaning, if a set of manual tuned parameters for image capture is to be used for training a custom object detection model in an industrial practice, the same capture settings should be utilized for detection. The laser welds were not detected by the trained model utilizing the Zivid camera. Captures of the welds were taken with a mobile phone camera, and with a closer distance compared to Zivid's operating range. Some debatable points on why the algorithm did not recognize the weld: hue differentiation and labeling coordinates/-format and augmentation techniques. The images captured with the mobile phone camera compared to the captures from Zivid camera differentiates in colors distributed in each pixel because of different acquisition settings. Weld labels were extracted from polygon annotations with x- and y values from the image frame of mobile phone captures. When the pixel coordinates for the weld and image size in pixels for training and detection are compared, the label to image ratio are quite noticeable. Meaning with a capture from a short distance (e.g. the mobile phone camera), x- and y coordinates has larger displacement between each coordinate when labeling, compared to a photo taken from a greater distance. It would be reasonable to utilize the same camera for training captures and detection captures. Dataset 1 was not exposed to augmentation techniques such as saturation, brightness and exposure. By adding images in the dataset with these augmentation techniques, hue differentiation from mobile camera and Zivid camera could have been improved. Detection of classes in Dataset 1 (specially the laser weld) could be improved if some of these measures was accounted: more data should be included and annotated, and more augmentation techniques should be applied. It could also be interesting to see a comparison between a dataset labelled with bounding box format and one with polygon labeling.

As mentioned in Chapter 7 Dataset 2 was generated by a short submission deadline. The results were not defensible, and motivates for further improvement. mAP did not rise, even with an increase of epochs. The train-val-test split compared to Roboflows recommendation was reasonable. The exported dataset including only 58 images. A polygon labeling format for a circular shape includes a lot of vertices, which results in a lot more data to go through training. From these points, a further improvement of Dataset 2 would be to enlarge the amount of data/images.

Chapter 9

Conclusion & Further work

Conclusion

Automatic module dismantling concepts have been evaluated for a Volkswagen E-Golf 2019 battery module. The presence of irreversible fasteners, motivated destructive dismantling techniques. The hydraulic actuated concept was investigated to separate the welded compressive plates within the casing. A FEM analysis with different configurations was performed to evaluate the most effective hydraulic solution when analysing the Von Mises stress. An automated milling concept has been presented as well, manipulators from ABB have been assessed and the feasibility has been verified based on results from manual milling operation. The suggested dismantling process was developed by merging robotic control and computer vision in a system architecture. The experimental setup included an ABB IRB4400 industrial robot which was controlled by open source software based on Robot Operating System (ROS) and MoveIt, while the computer vision system included a cutting-edge 3D camera, Zivid, and the best-suited object detection algorithm, YOLOv5. In order to accomplish the best suited capture configurations adjustable acquisition settings were evaluated with scientific methods. Custom object detection models which recognized components from the BM were trained and tested with image captures. Overall, this research shows that even though battery modules have irreversible fasteners, they can be dismantled automatically. Hence, it is possible to automate a disassembly process for the purpose of recycling. However, the compressive plates from the battery module motivated for a dismantling concept utilizing cutting tools. The proposed approaches were tested on a single battery module (Egolf 2019), but they are adaptable enough to be applied to a wide range of EV battery modules.

Further work

The generic tool and technology in this study is primarily developed with respect to the Battery Module from a Volkswagen E-Golf (2019). It should also be tested with a large variety of other battery modules with different geometries and connectors. The generic tool and technology developed in this study will be part of the toolbox used to dismantle any type of modules. Disassembly sequences will be automatically generated based on AI with support from the perception system and different dismantling operations developed in this thesis.

`abb_robot_driver`

Near the end of the project, a driver for the ABB robot which supports Ubuntu Focal was discovered [32]. ROS Noetic is the supported ROS distribution for Ubuntu Focal. ROS Melodic differs from ROS Noetic in terms of python versions. As mentioned in 4.5, dependencies in ROS Melodic are primarily built with Python 2.7. I.e. the `abb_driver` for robotic control operates on Python 2.7. State-of-the-art object detectors like YOLOv5 with

its requirements utilize Python 3. Different Python versions does not communicate. Hence, ROS dependencies like MoveIt must be executed separately. However, Python 2.7 scripts can be called from Python 3 by removing the system path of ROS-Python dependencies and specify Python 3 in the shebang as seen below.

```
#!/usr/bin/env python3

import sys

sys.path.remove('/opt/ros/melodic/lib/python2.7/dist-packages')
```

By implementing ROS Noetic, seamless integration between ABB, Zivid and Python 3 dependencies could be achieved. The proposed system architecture for future academic research can be seen in Fig. 9.1. If to be compared with Fig. 4.3, the Python 3 branch would be separate from ROS Melodic.

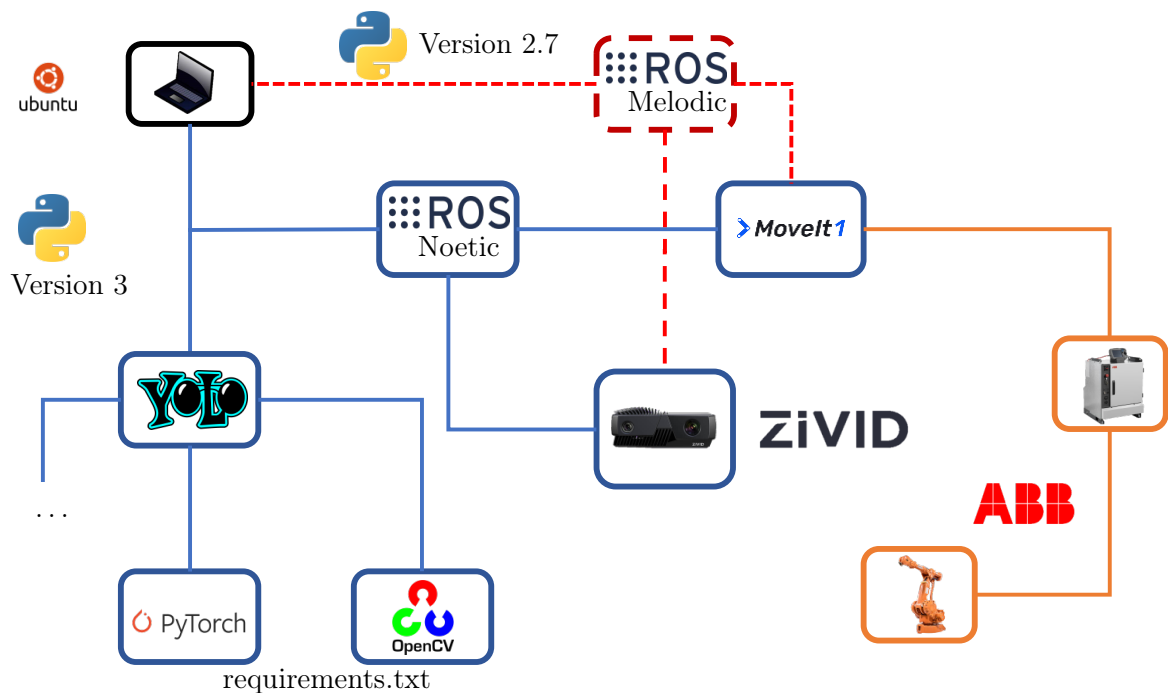
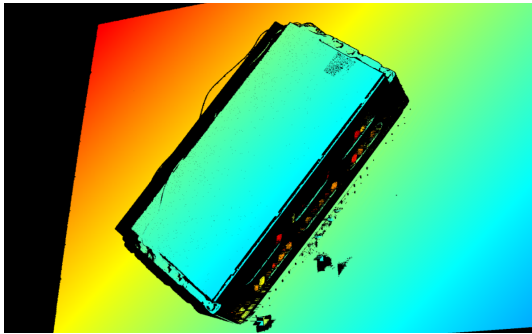


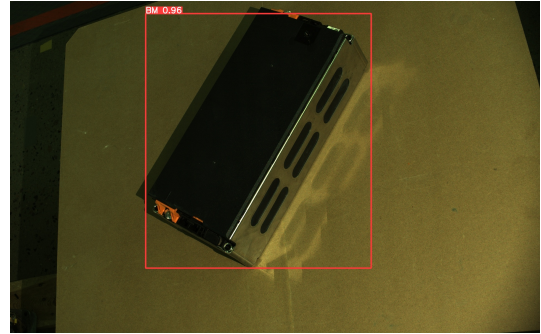
Figure 9.1: System architecture, ROS Noetic

Pose estimation

With the detected components, it would be necessary to obtain the pose of the object with respect to the world frame in order to process any automatic dismantling concept. By merging point cloud data generated from structured light, and the detected object in 2D with YOLO, the pose can be obtained. In order to read the point cloud, a point cloud viewer must be installed, e.g. [point cloud library \(pcl\)](#). The intrinsic- and extrinsic matrices from the camera must be calculated to determine the pose of an object. The overall concept is that the center of a given class from the bounding box' in x- and y- coordinates is extracted, and then merged with the distance acquired from the point cloud [11].



(a) Distance obtained from point cloud,
Z-value cloud



(b) X- and Y-center pixel coordinate from
bounding box

Figure 9.2: Merge of point cloud and 2D image for pose estimation

Appendix A

ABB IRB 4400 datasheet

ROBOTICS

Product specification

IRB 4400



Trace back information:
Workspace 21A version a11
Checked in 2021-03-19
Skribenta version 5.4.005

Product specification

**IRB 4400/60
IRB 4400/L10**

Document ID: 3HAC042478-001

Revision: P

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damage to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

© Copyright 2004-2021 ABB. All rights reserved.
Specifications subject to change without notice.

Table of contents

Overview of this product specification	7
1 Description	9
1.1 Structure	9
1.1.1 Introduction to structure	9
1.1.2 Different robot versions	11
1.2 Standards	15
1.2.1 Applicable standards	15
1.3 Installation	17
1.3.1 Introduction to installation	17
1.3.2 Operating requirements	18
1.3.3 Mounting the manipulator	19
1.4 Calibration and references	22
1.4.1 Calibration methods	22
1.4.2 Fine calibration	24
1.4.3 Absolute Accuracy calibration	25
1.5 Load diagrams	27
1.5.1 Introduction to load diagrams	27
1.5.2 Maximum load and moment of inertia for full and limited axis 5 movement	30
1.5.3 Wrist torque	31
1.6 Mounting equipment	32
1.7 Maintenance and troubleshooting	35
1.8 Robot motion	36
1.9 Signals	40
2 Specification of variants and options	41
2.1 Introduction to variants and options	41
2.2 Manipulator	42
2.3 Positioners	47
2.4 Track motion	48
2.5 Floor cables	49
2.5.1 Manipulator	49
2.6 User documentation	50
3 Accessories	51
3.1 Introduction to accessories	51
Index	53

This page is intentionally left blank

Overview of this product specification

About this product specification

It describes the performance of the manipulator or a complete family of manipulators in terms of:

- The structure and dimensions prints
- The fulfillment of standards, safety and operating requirements
- The load diagrams, mounting or extra equipment, the motion and the robot reach
- The specification of variants and options available

Usage

Product specifications are used to find data and performance about the product, for example to decide which product to buy. How to handle the product is described in the product manual.

Users

It is intended for:

- Product managers and product personnel
- Sales and marketing personnel
- Order and customer service personnel

References

Reference	Document ID
<i>Product specification - Controller IRC5</i> IRC5 with main computer DSQC1000.	3HAC047400-001
<i>Product specification - Controller software IRC5</i> IRC5 with main computer DSQC1000 and RobotWare 5.6x.	3HAC050945-001
<i>Product specification - Controller software IRC5</i> IRC5 with main computer DSQC1000 and RobotWare 6.	3HAC050945-001
<i>Product manual - IRB 4400</i>	3HAC022032-001
<i>Product specification - Robot user documentation, IRC5 with RobotWare 6</i>	3HAC052355-001

Revisions

Revision	Description
-	<ul style="list-style-type: none"> • Replaces 3HAC9117-1 (English), 3HAC10768-1 (French), 3HAC10395-1 (German), 3HAC10782-1 (Spanish), and 3HAC10761-1 (Italian) • Table for ambient temperature adjusted • Text for Foundy Prime updated
A	<ul style="list-style-type: none"> • Machinery directive updated
B	<ul style="list-style-type: none"> • General updates and minor corrections

Continues on next page

Overview of this product specification

Continued

Revision	Description
C	<ul style="list-style-type: none">• General updates and minor corrections• Option Foundry Prime removed• Figure for mounting equipment changed
D	<ul style="list-style-type: none">• Text for ISO test adjusted• IRB 4400/L10 added
E	<ul style="list-style-type: none">• Minor updates/corrections
F	<ul style="list-style-type: none">• Text for Foundry Plus updated.
G	<ul style="list-style-type: none">• Drawing main dimension for IRB 4400/L10 updated
H	<ul style="list-style-type: none">• Minor corrections/updates
J	<ul style="list-style-type: none">• RT and AT adjusted after new ISO measurement
K	<ul style="list-style-type: none">• Values changed, see Performance according to ISO 9283 on page 38• Screw size of IRB 4400 changed to M20• Restriction of load diagram added
L	<ul style="list-style-type: none">• Updated list of applicable standards.• Drawing view of mounting holes for extra equipment added.
M	Published in release R18.1. The following updates are done in this revision: <ul style="list-style-type: none">• Change the general description of load diagrams.
N	Published in release R20D. The following updates are done in this revision: <ul style="list-style-type: none">• Warranty section updated.• Updated information about <i>Absolute Accuracy</i>.
P	Published in release R21A. The following updates are done in this revision: <ul style="list-style-type: none">• Updated repeatability values.

1 Description

1.1 Structure

1.1.1 Introduction to structure

Robot family

The IRB 4400 is a 6-axis industrial robot, designed specifically for manufacturing industries that use flexible robot-based automation. The robot has built-in process ware, an open structure that is specially adapted for flexible use, and can communicate extensively with external systems.

Operating system

The robot is equipped with the IRC5 controller and robot control software, RobotWare. RobotWare supports every aspect of the robot system, such as motion control, development and execution of application programs, communication etc. see Product specification - Controller IRC5 with FlexPendant.

Safety

Safety standards valid for complete robot, manipulator and controller.

Additional functionality

For additional functionality, the robot can be equipped with optional software for application support - for example gluing and welding, communication features - network communication - and advanced functions such as multi-tasking, sensor control, etc. For a complete description on optional software, see Product specification - Controller software IRC5.

Foundry Plus**Foundry Plus**

The Foundry Plus option is designed for harsh environments where the robot is exposed to sprays of coolants, lubricants and metal spits that are typical for die casting applications or other similar applications. Typical applications are spraying insertion and part extraction of die-casting machines, handling in sand casting and gravity casting, etc. (Please refer to Foundry Prime for washing applications or other similar applications). Special care must be taken in regard to operational and maintenance requirements for applications in foundry are as well as in other applications areas. Please contact ABB Robotics Sales organization if in doubt regarding specific application feasibility for the Foundry Plus robot. The Foundry Plus robot is painted with two-component epoxy on top of a primer for corrosion protection. To further improve the protection additional preventive measures are applied to exposed and crucial areas. Continuous splashing of water or other similar rust formation fluids may cause corrosion on the robots unpainted areas, joints, or other unprotected surfaces. Under these circumstances it is recommended to add rust inhibitor to the fluid or take other measures to prevent potential rust formation. The entire robot is IP67 compliant according to IEC 60529 - from base

Continues on next page

1 Description

1.1.1 Introduction to structure

Continued

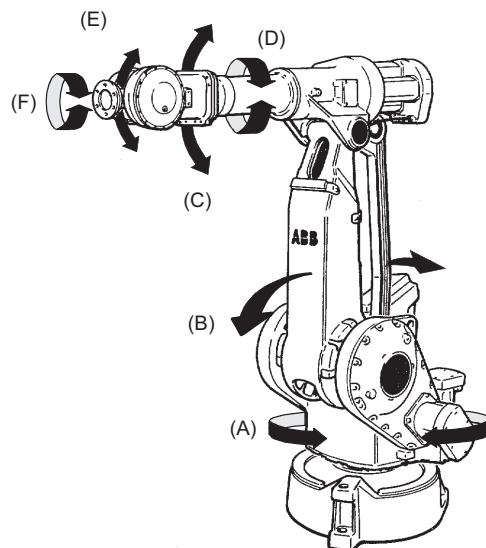
to wrist, which means that the electrical compartments are sealed against water and solid contaminants. Among other things all sensitive parts are better protected than the standard offer.

Selected Foundry Plus features:

- Improved sealing to prevent penetration into cavities to secure IP67
- Additional protection of cabling and electronics
- Special covers protecting cavities
- Well-proven connectors
- Rust preventives on screws, washers and unpainted/machined surfaces
- Extended service and maintenance program

The Foundry Plus robot can be cleaned with appropriate washing equipment according to product manual. Appropriate cleaning and maintenance are required to maintain the Foundry Plus protection, for example can rust preventive be washed off with wrong cleaning method.

Manipulator axes



xx110000607

Pos	Description	Pos	Description
A	Axis 1	B	Axis 2
C	Axis 3	D	Axis 4
E	Axis 5	F	Axis 6

1.1.2 Different robot versions

General

The IRB 4400 is available in two variants, to be floor mounted (no tilting allowed around X-axis or Y-axis).

Robot type	Handling capacity (kg)	Reach (m)
IRB 4400/60	60	1.96
IRB 4400/L10	10	2.55

Manipulator weight

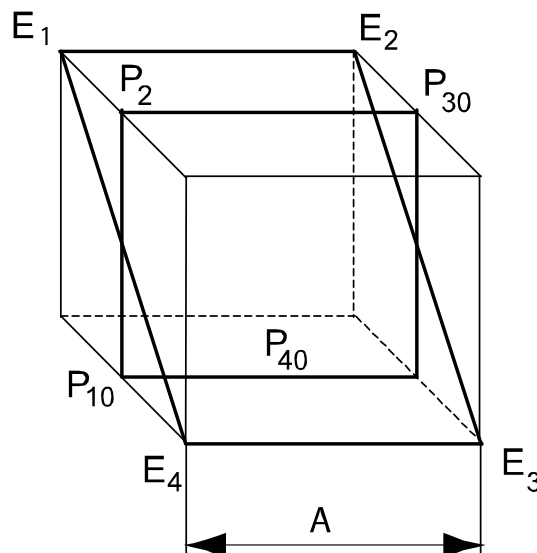
Robot type	Weight
IRB 4400/60	1040 kg
IRB 4400/L10	1040 kg

Other technical data

Data	Description	Note
Airborn noise level	The sound pressure level outside the working space	< 70 dB (A) Leq (acc. to Machinery directive 2006/42/EG)

Power consumption at max load

Type of Movement	IRB 4400/60	IRB 4400/L10
ISO Cube Max. velocity	1.33 kW	1.28 kW



xx0900001012

Pos	Description
A	630 mm ⁱ

ⁱ 1000 mm valid for IRB 4400/L10

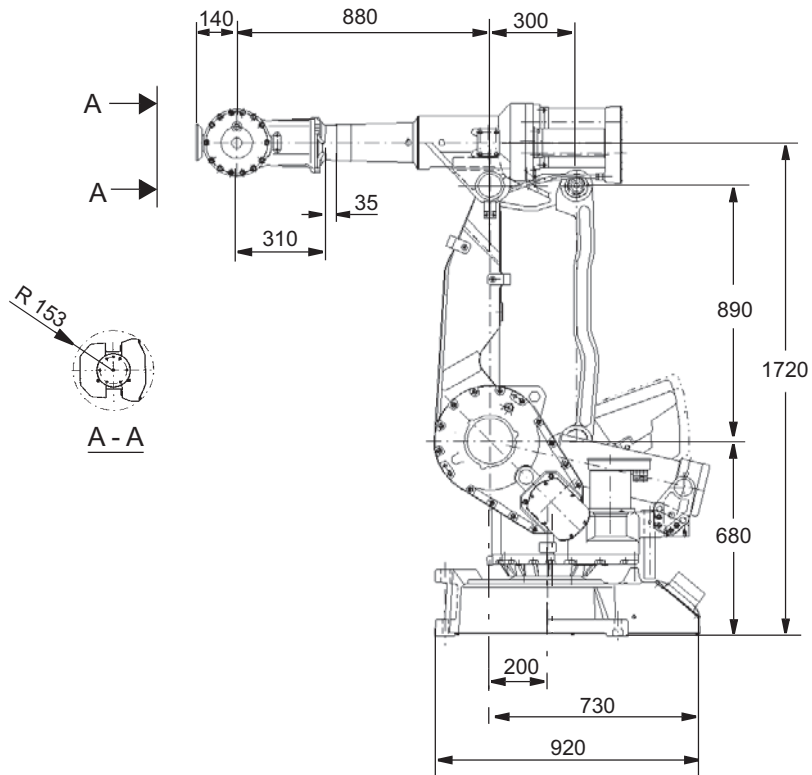
Continues on next page

1 Description

1.1.2 Different robot versions

Continued

Dimensions IRB 4400/60

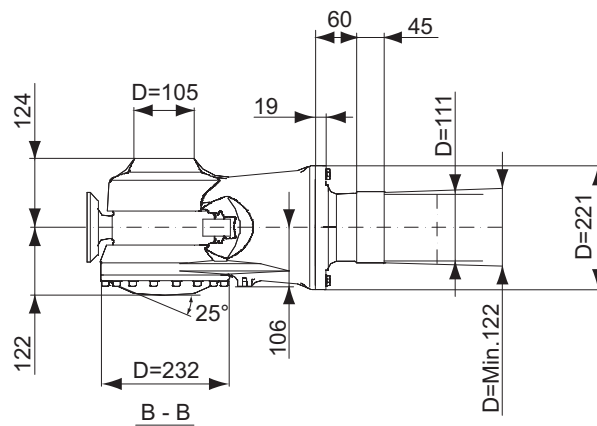
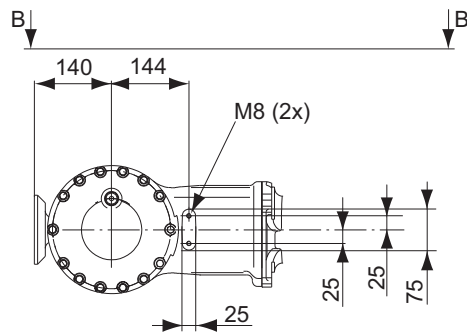


xx1100000598

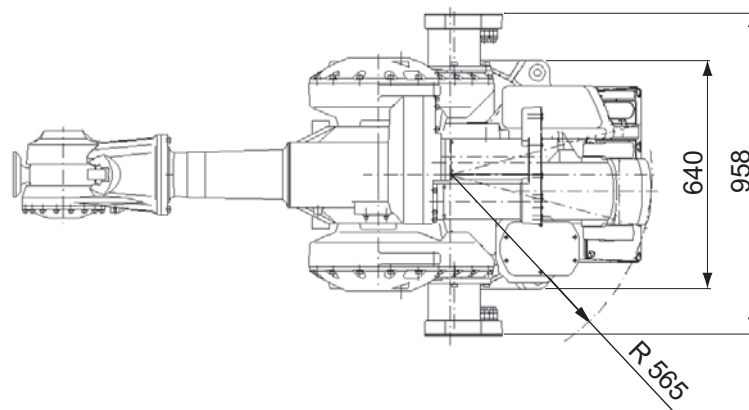
Continues on next page

1 Description

1.1.2 Different robot versions Continued



xx110000608



xx110000599

Continues on next page

Appendix B

ABB IRBT 4004 datasheet

IRBT 4004/6004/7004 Track Motions for robots

The IRBT X004 from ABB is the only track motion platform on the market to guarantee high speed, precision accuracy, and great flexibility.



Outstanding speed and accuracy

As first on the market, ABB's track motions and its respective robot is a seven-axis dynamic model. ABB's unique Quick-Move and TrueMove can be fully exploited, which means optimal movement for the robot and the track with actual load. Furthermore, path accuracy and speed are optimized.

The speed position-to-position, which is the real benchmark of the capability of the track, has improved by more than 40% compared to earlier models.

Benefits

- Path accuracy best in class
- Simple, robust and compact design
 - Less parts, many common modules for complete range
 - Easy to adjust at installation and start up
 - The same mechanical footprint of the track for all robot models
- Improved performance
 - Shorter cycle time
 - Improved path accuracy
- Robot position chosen at installation
- Wide range of options for different applications

Options

- Mirrored version
- Double carriages
- Pedestals
- Customer Signal and power cable (spot welding, material handling, etc)
- Lubrication (standard = manual lubrication)
 - Prepared for local adaptation (no pump included)
 - Central lubrication system incl pump
- Foundry
 - Travel length from 1.9 to max 7.9 for IRBT 4004
 - From 1.7 to max 7.7 for IRBT 6004 and 7004
- Cable arrangement

	Robot	Travel length	No of robots	Mounting pos
IRBT 4004	IRB 4400/ 4450S/4600	1.9... 19.9 m/1 m step	One or two/track	Floor
IRBT 6004/ 7004	IRB 6620/6650S/6700	1.7... 19.7 m/1 m step	One or two/track	Floor
IRBT 7004	IRB 7600	1.7... 19.7 m/1 m step	One or two/track	Floor

Cable arrangement

Plastic with cover - standard

Pos to Pos time (s) *)	1 m	2 m	3 m	4 m	5 m
IRBT 4004	< 1.2	< 1.7	< 2.2	< 2.7	< 3.2
IRBT 6004	< 1.5	< 2.1	< 2.8	< 3.4	< 4.0
IRBT 7004	< 1.7	< 2.6	< 3.4	< 4.2	< 5.0

*) With max load

Acceleration/Retardation (m/s²) X004

IRB 4004	2.5*
IRB 6004	2.0*
IRB 7004	1.8*

* Dep. on actual load

Speed (m/s)

IRB 4004	2.0
IRB 6004	1.6
IRB 7004	1.2

For more information please contact:

ABB AB

Robotics

Hydrovägen 10

SE-721 36 Västerås, Sweden

Phone: +46 21 325000

www.abb.com/robotics

Note

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents - in whole or in parts - is forbidden without prior written consent of ABB.

Copyright© 2016

ABB All rights reserved

Appendix C

Zivid One+ datasheet

Zivid One⁺

Technical specification

Zivid One+ S (ZVD1P-S)

Zivid One+ M (ZVD1P-M)

Zivid One+ L (ZVD1P-L)

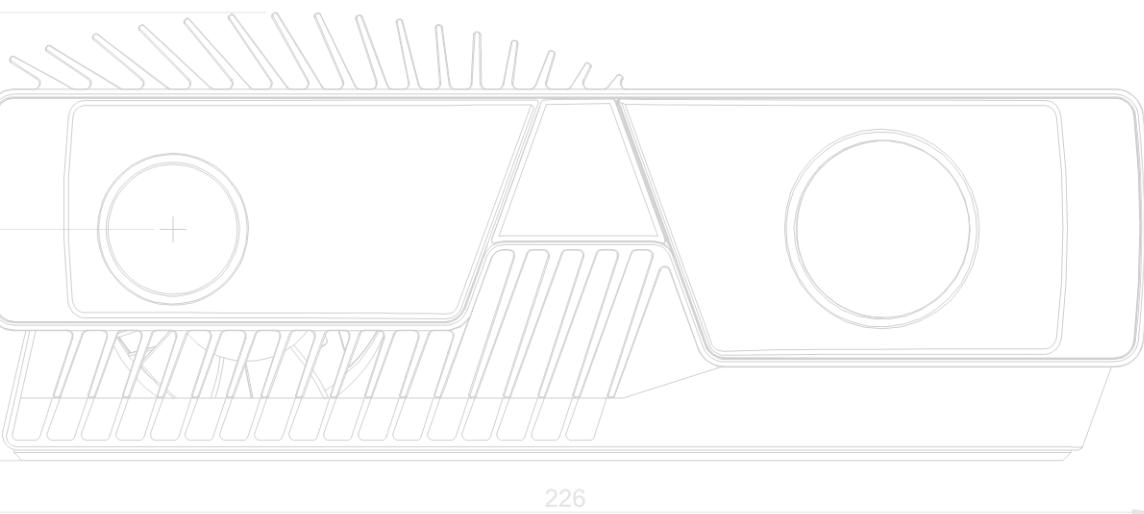


Table of Contents

- Table of Contents..... 2
- General specifications..... 3
- Operating distance and field of view..... 4
- Accuracy specifications 8
 - Common conditions.....8
 - Zivid One+ S Typical Specifications 9
 - Zivid One+ M Typical Specifications 12
 - Zivid One+ L Typical Specifications 15
- Physical specifications..... 18
- Mechanical drawings 19
 - Connectors**.....20
- Revision history 21

General specifications

Model (Part number)	Zivid One+ S (ZVD1P-S) Zivid One+ M (ZVD1P-M) Zivid One+ L (ZVD1P-L)
3D technology	Structured light
Imaging	1920 x 1200 (2.3Mpixel) Native 3D Color
Point cloud output	3D (XYZ) + Color (RGB) + SNR
Exposure time (minimum per pattern projection)	6.500 ms
Aperture (A)	f/1.4 to f/32
Gain (G)	1x to 16x
Projector Brightness (B)	0.25x to 1.8x 1x = 400 lumens
Calibration	Factory calibrated
Safety and EMC	CE CB EN60950 FCC Class A
Typical capture time ¹	100 ms to 1 s

¹ From capture initialized until point cloud is ready to copy. Includes processing. Acquisition time can be shorter.

Operating distance and field of view

	S	M	L
Focus distance (mm)	500	1000	1800
Optimal working distance (mm)	350 to 700	700 to 1500	1200 to 2600
Recommended working distance (mm)	300 to 1000	500 to 2000	1200 to 3000
Field of view (mm)	164 x 132 at 300 350 x 220 at 500 621 x 439 at 1000	433 x 271 at 600 702 x 432 at 1000 1330 x 871 at 2000	843 x 530 at 1200 1252 x 783 at 1800 2069 x 1310 at 3000
Spatial resolution (mm)	0.18 at 500 4.00×10^{-4} per distance (z) in mm	0.37 at 1000 3.71×10^{-4} per distance (z) in mm	0.67 at 1800 3.67×10^{-4} per distance (z) in mm

Figure 1 - Zivid One+ S FOV

All values in degrees or mm.

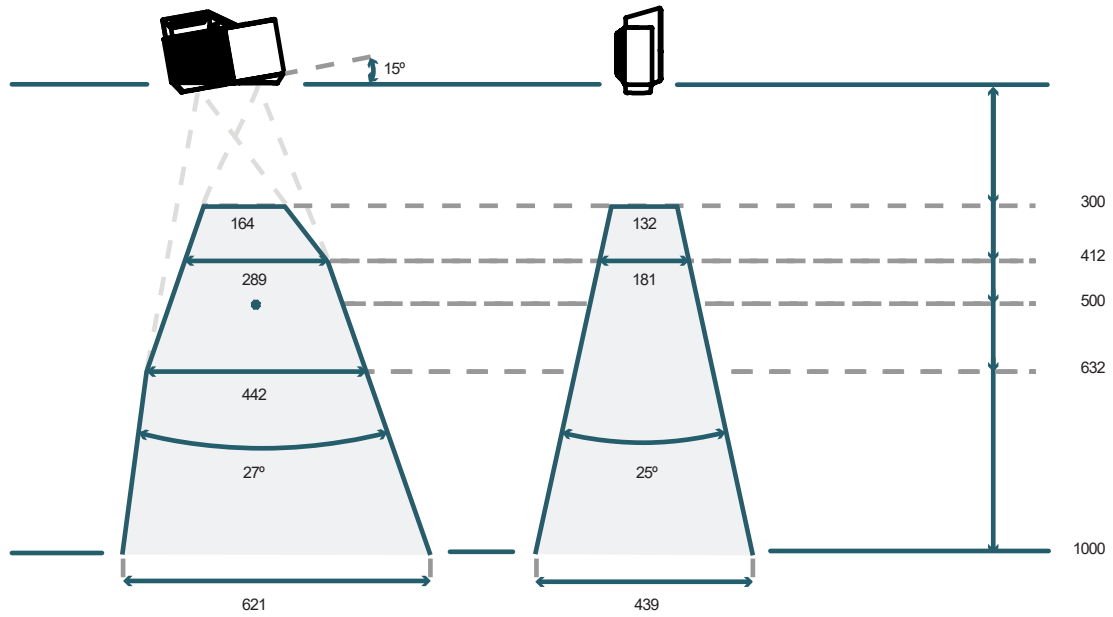


FIGURE 2 - ZIVID ONE+ S SPATIAL RESOLUTION VS. DISTANCE

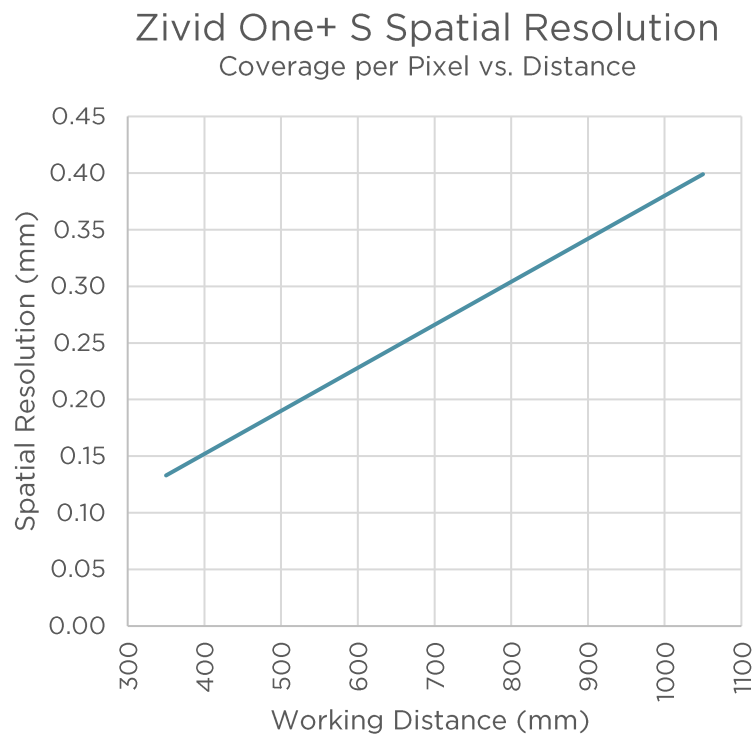


Figure 3 - Zivid One+ M FOV

All values in degrees or mm.

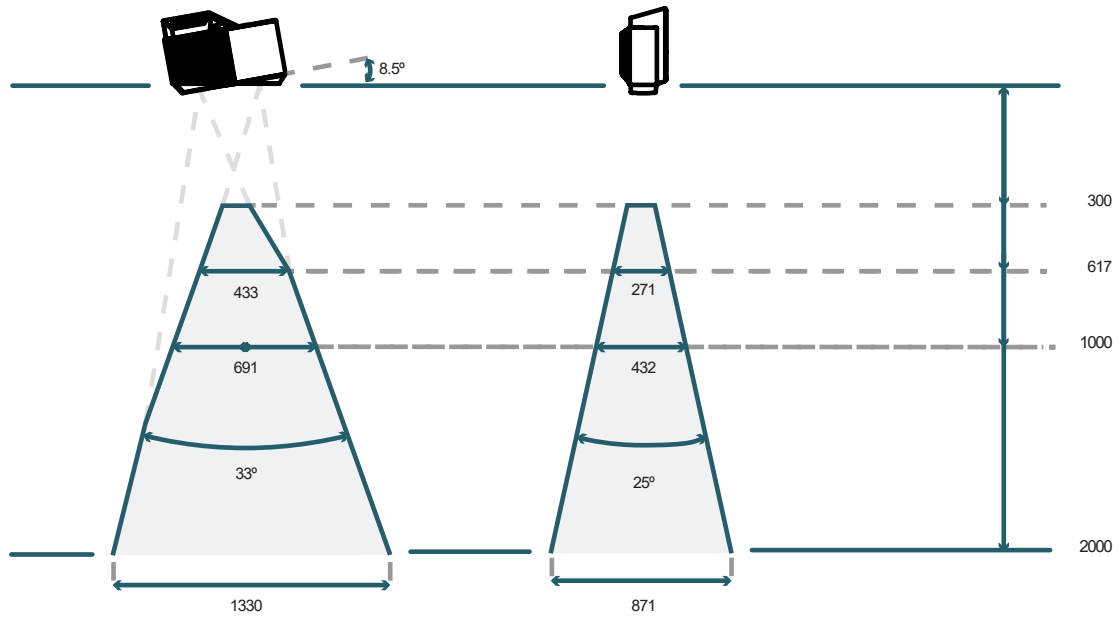


FIGURE 4 - ZIVID ONE+ M SPATIAL RESOLUTION VS. DISTANCE

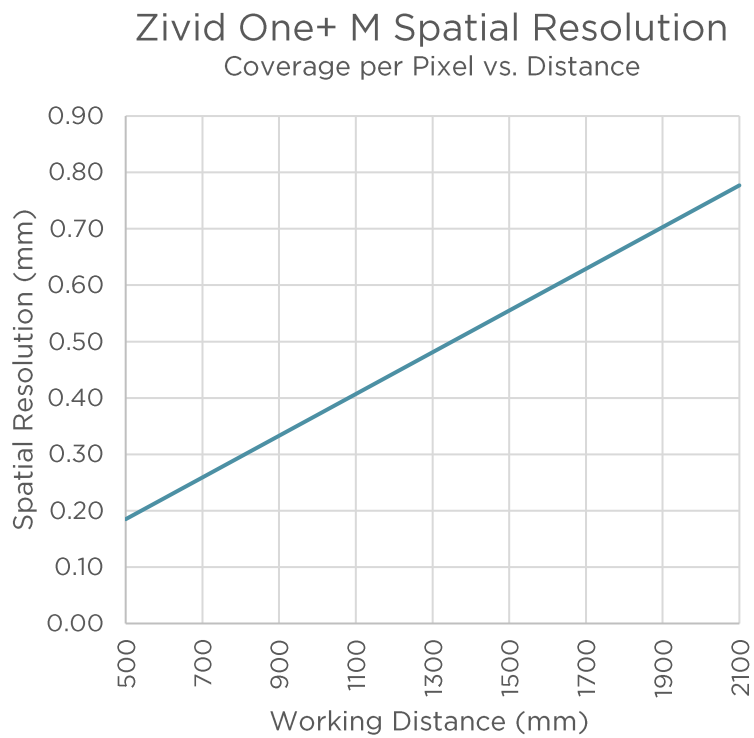


Figure 5 - Zivid One+ L FOV

All values in degrees or mm.

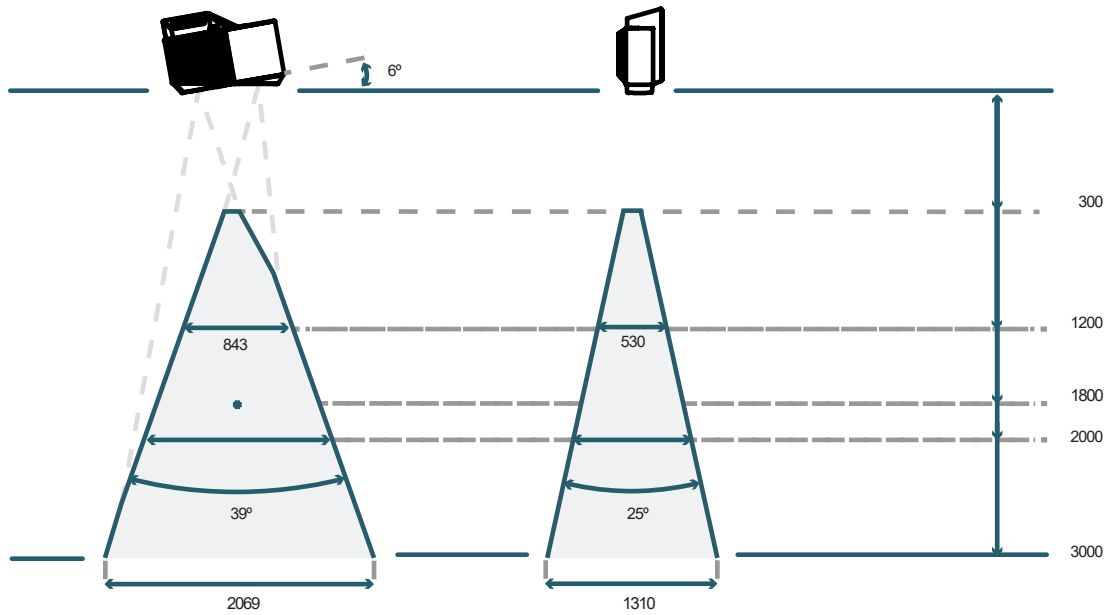
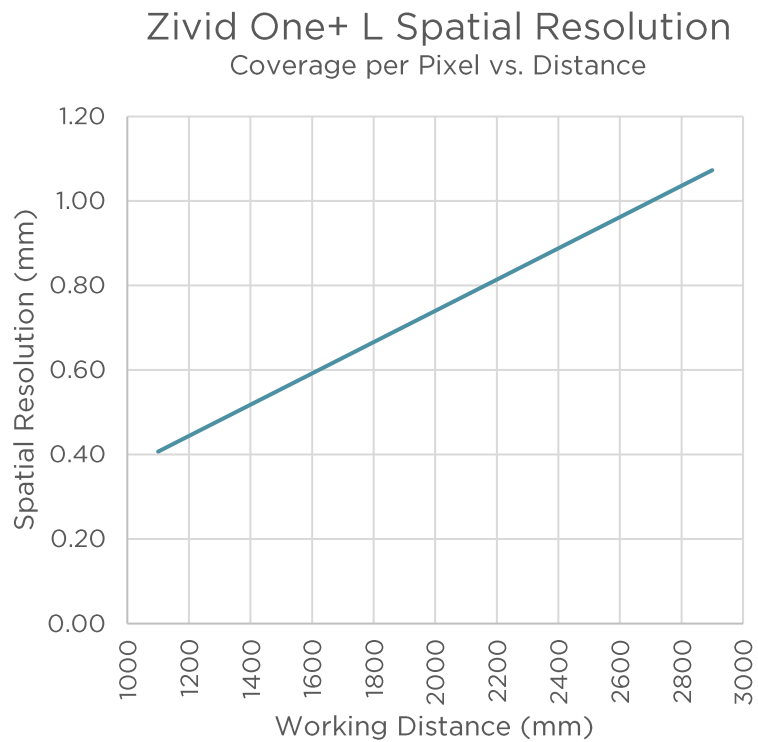


FIGURE 6 - ZIVID ONE+ L SPATIAL RESOLUTION VS. DISTANCE



Accuracy specifications

Common conditions

The following table outlines the conditions applied under test and to all specifications unless otherwise stated.

Parameter	Description	Typical
Working distance (D)		Zivid One+ S: 500 mm
	Focus distance	Zivid One+ M: 1000 mm
		Zivid One+ L: 1800 mm
Working distance (D)		Zivid One+ S: 350 - 700 mm
	Optimal working distance	Zivid One+ M: 700 - 1500 mm
		Zivid One+ L: 1300 - 2600 mm
Ambient temperature (Ta)	Typical temperature	15 - 30 °C
	Full temperature range	10 - 40 °C
Ambient light (La)		0 lux
Aperture (A)		f/8.0 - f/2.0
Gain (G)		1.0x
Projector Brightness (B)		1 - 1.8 x
Capture time	Acquisition time used during measurement	> 85 ms
	Capture time used during measurement	> 200 ms
Duty Cycle	Capture-to-Idle time ratio	5 - 30 %
Other		81% center crop (90% × 90%)
		HDR = off
		10 min warm-up
		Applied in-field correction

Zivid One+ S Typical Specifications

Typical numbers are given at common conditions unless otherwise specified.

Property	Description	Typical
Warm-up time	Minimum recommended time needed for camera to stabilize from an idle state assuming capturing at a constant rate. Some trueness changes may be experienced during warm-up phase.	10 minutes
Point precision	1σ Euclidian distance variation for a point between consecutive measurements at focus distance, D. ²	25 μm
Local Planarity Precision	1σ Euclidian distance variation from a plane for a set of points within a smaller local region at focus distance, D. ²	40 μm
Global Planarity Trueness	Average deviation from a plane in field of view at focus distance, D.	< 100 μm
Dimension Trueness	70-percentile dimension error in field of view at focus distance, D, and typical temperature range.	< 0.15 %
	70-percentile dimension error in field of view within optimal working distance and typical temperature range.	< 0.20 %
	70-percentile dimension error in field of view within optimal working distance and full temperature range.	< 0.30 %

² Measured with Gaussian filter disabled.

FIGURE 7 - TYPICAL ZIVID ONE+ S POINT PRECISION VS. DISTANCE

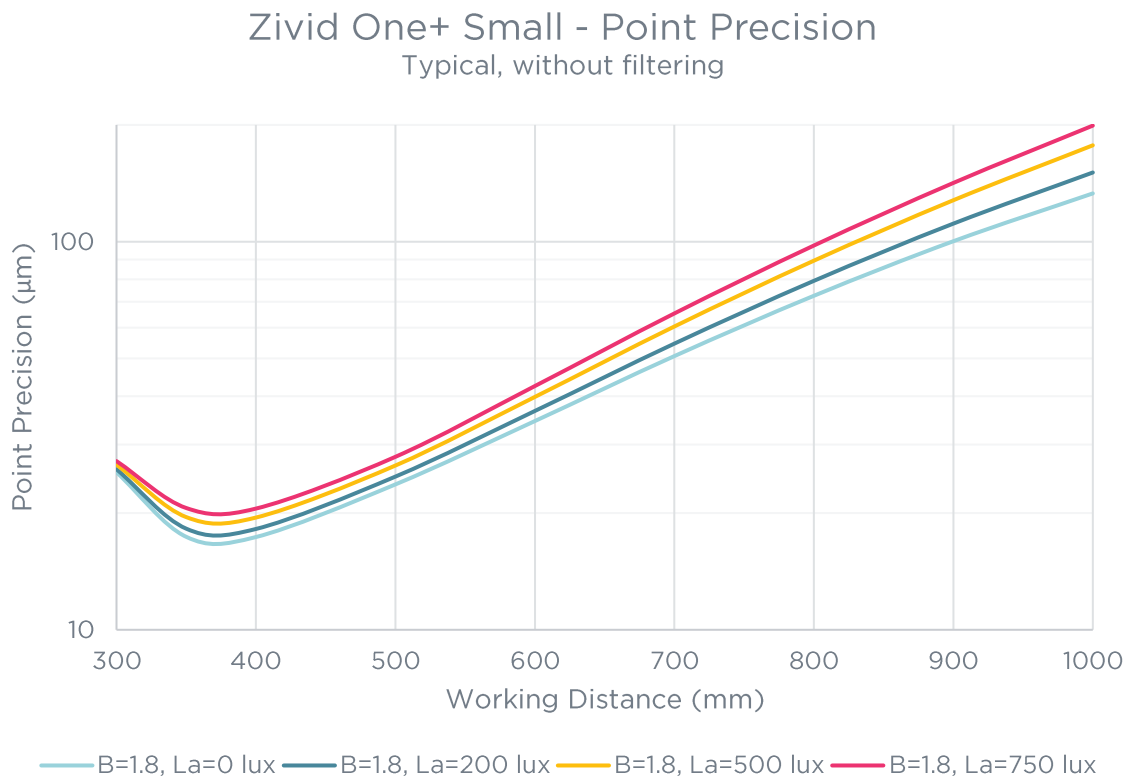


FIGURE 8 - TYPICAL ZIVID ONE+ S LOCAL PLANARITY PRECISION VS. DISTANCE

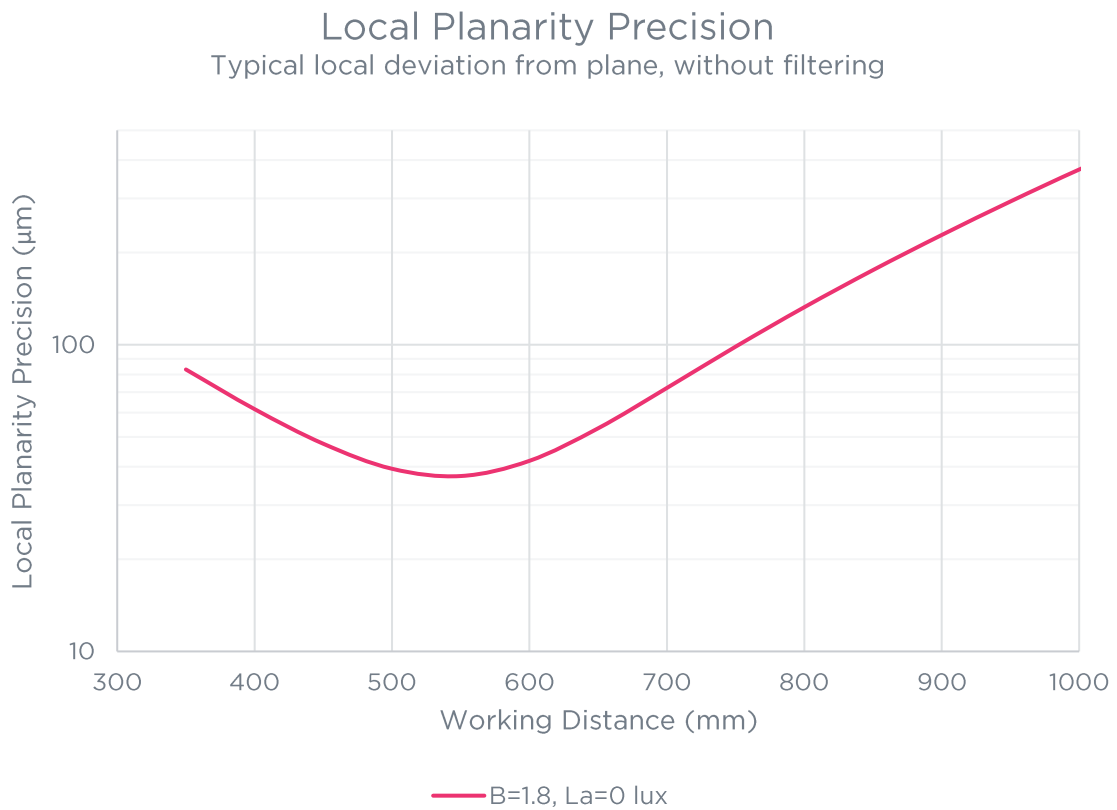


FIGURE 9 – TYPICAL ZIVID ONE+ S GLOBAL PLANARITY TRUENESS VS. DISTANCE

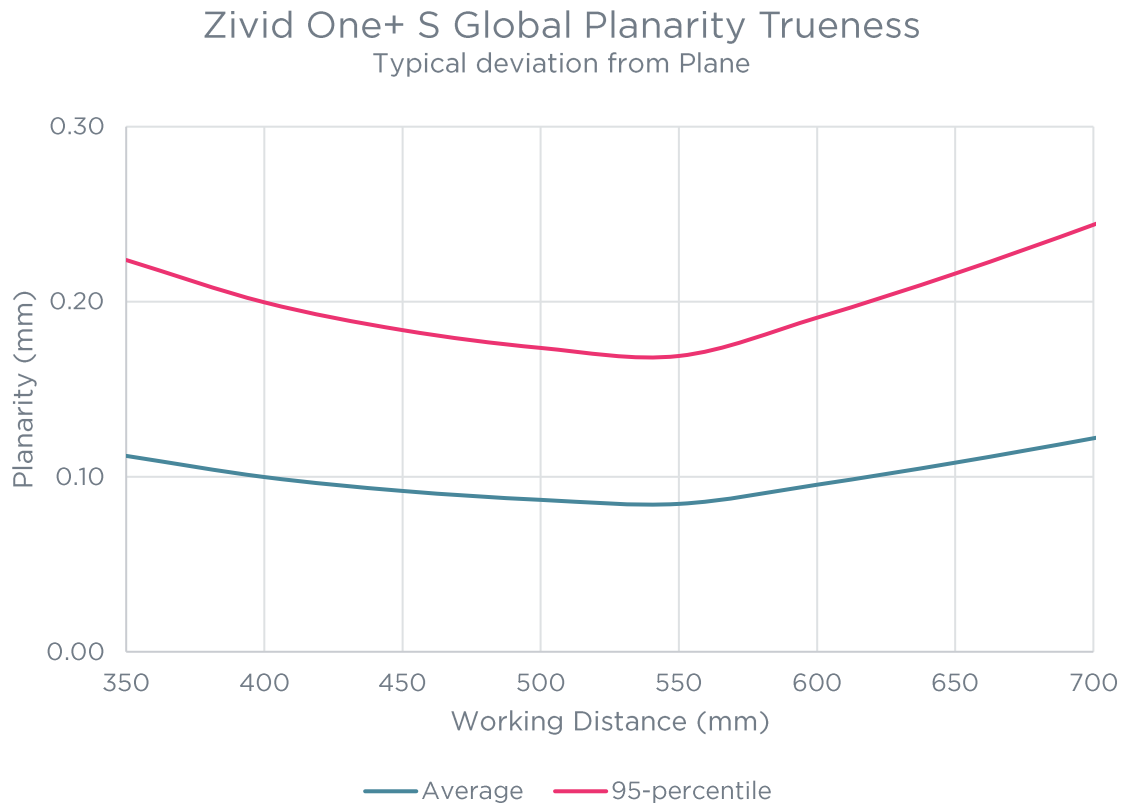
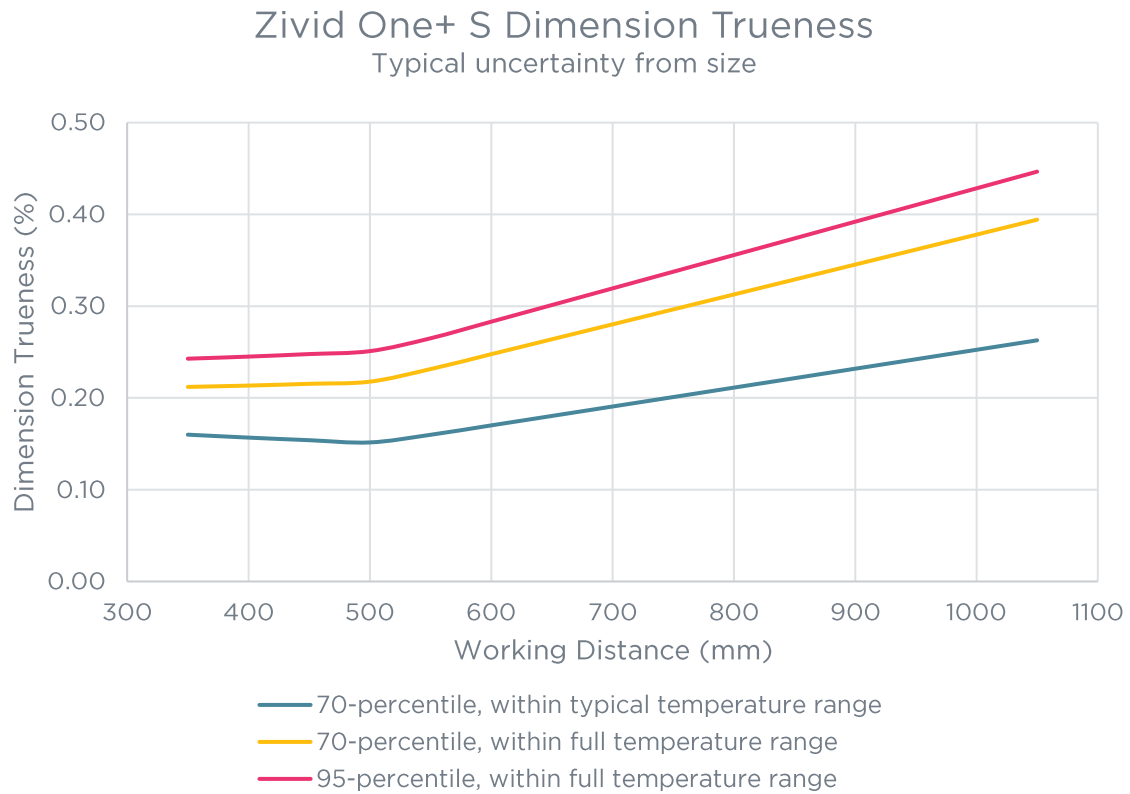


FIGURE 10 – TYPICAL ZIVID ONE+ S DIMENSION TRUENESS VS. DISTANCE



Zivid One+ M Typical Specifications

Typical numbers are given at common conditions unless otherwise specified.

Property	Description	Typical
Warm-up time	Minimum recommended time needed for camera to stabilize from an idle state assuming capturing at a constant rate.	10 minutes
	Some trueness changes may be experienced during warm-up phase.	
Point precision	1 σ Euclidian distance variation for a point between consecutive measurements at focus distance, D. ³	110 μ m
Local Planarity Precision	1 σ Euclidian distance variation from a plane for a set of points within a smaller local region at focus distance, D. ³	190 μ m
Global Planarity Trueness	Average deviation from a plane in field of view at focus distance, D.	< 100 μ m
Dimension Trueness	70-percentile dimension error in field of view at focus distance, D, and typical temperature range.	< 0.30 %
	70-percentile dimension error in field of view within optimal working distance and typical temperature range.	< 0.40 %
	70-percentile dimension error in field of view within optimal working distance and full temperature range.	< 0.50 %

³ Measured with Gaussian filter disabled.

FIGURE 11 – TYPICAL ZIVID ONE+ M POINT PRECISION VS. DISTANCE

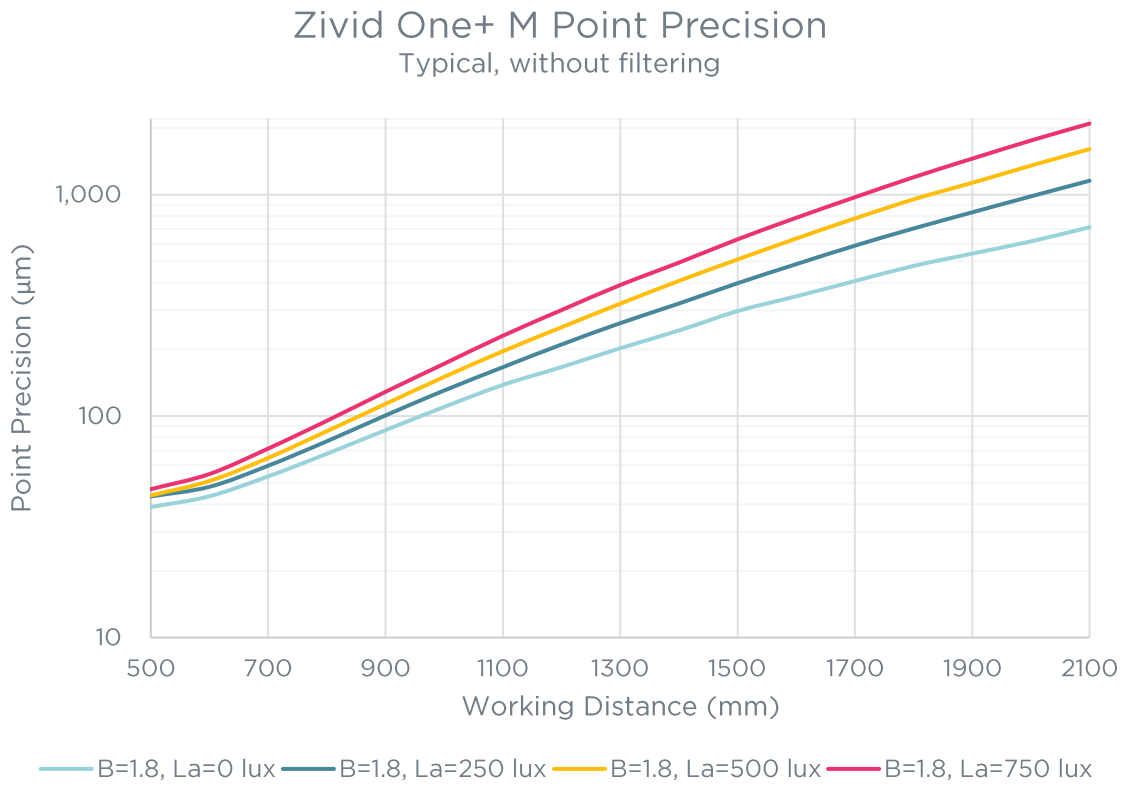


FIGURE 12 – TYPICAL ZIVID ONE+ M LOCAL PLANARITY PRECISION VS. DISTANCE

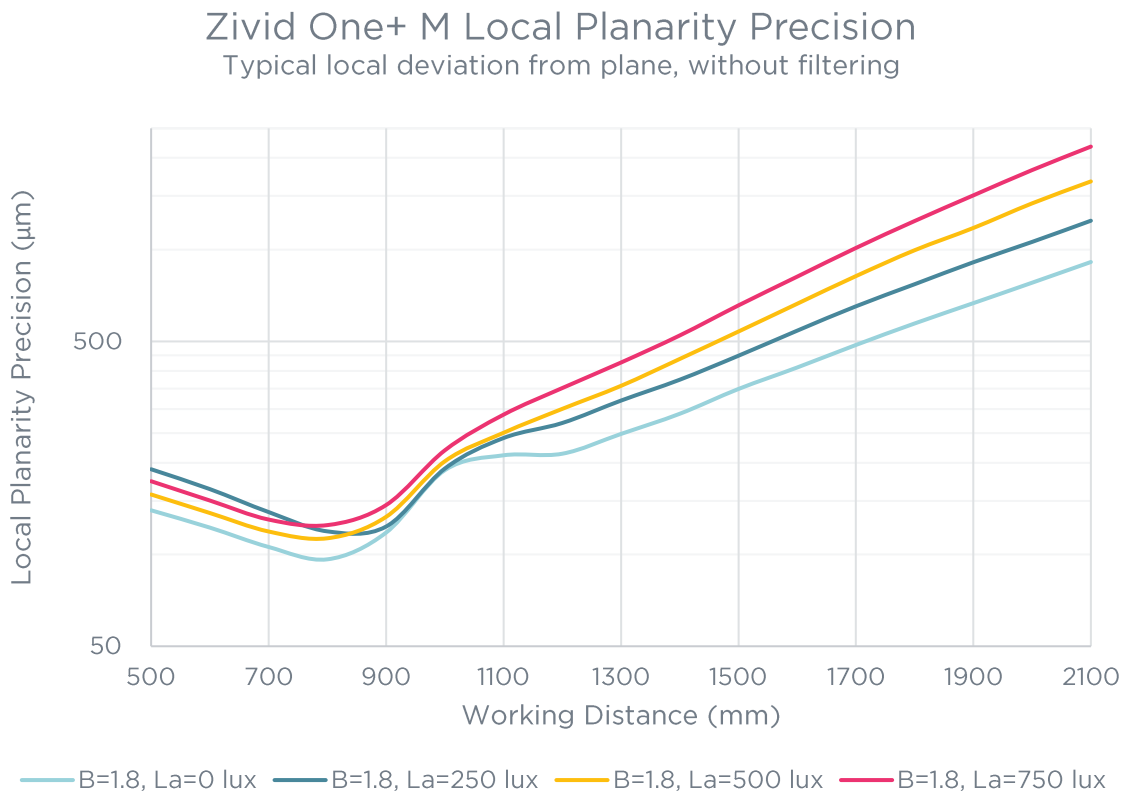


FIGURE 13 – TYPICAL ZIVID ONE+ M GLOBAL PLANARITY TRUENESS VS. DISTANCE

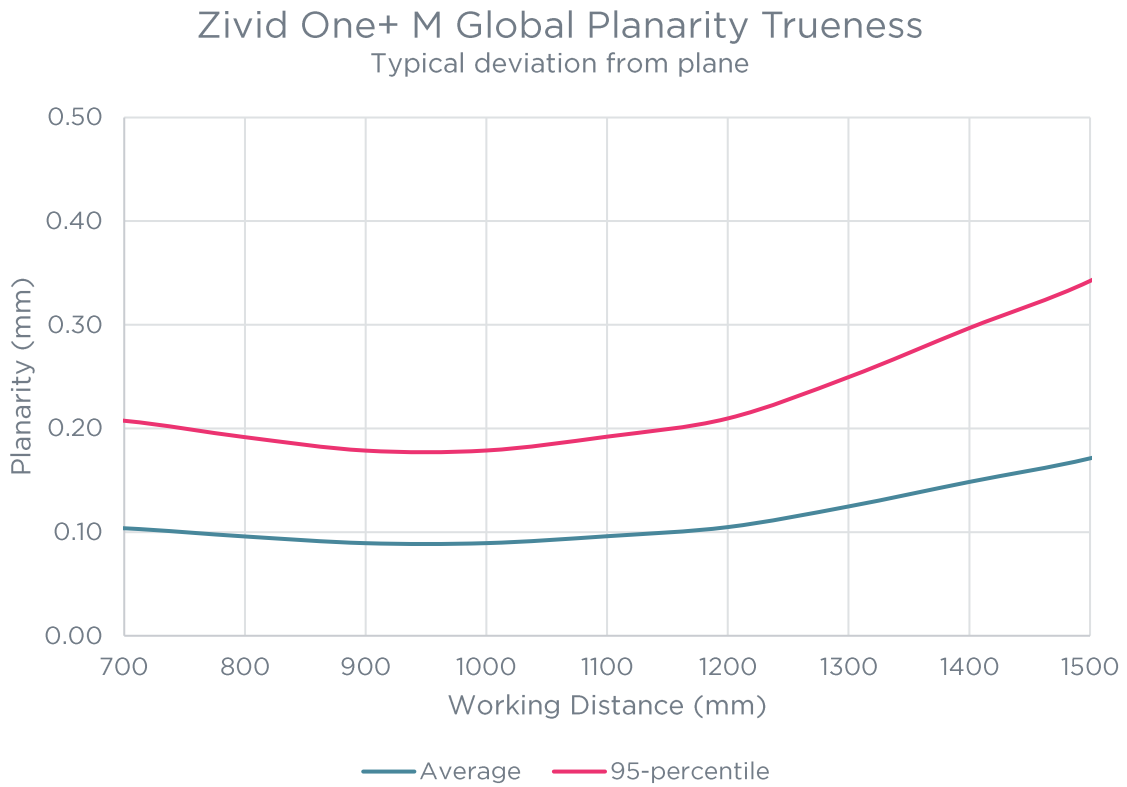
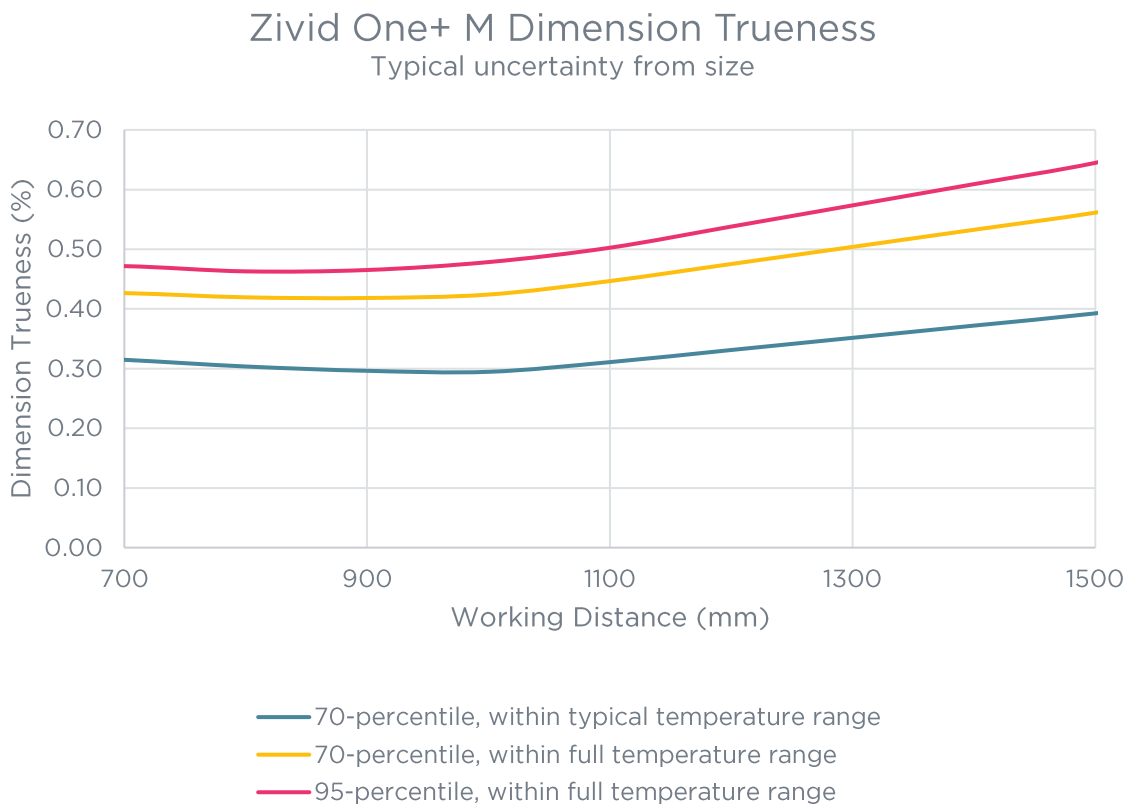


FIGURE 14 – TYPICAL ZIVID ONE+ M DIMENSION TRUENESS VS. DISTANCE



Zivid One+ L Typical Specifications

Typical numbers are given at common conditions unless otherwise specified.

Property	Description	Typical
Warm-up time	Minimum recommended time needed for camera to stabilize from an idle state assuming capturing at a constant rate. Some trueness changes may be experienced during warm-up phase.	10 minutes
Point precision	1σ Euclidian distance variation for a point between consecutive measurements at focus distance, D. ⁴	350 μm
Local Planarity Precision	1σ Euclidian distance variation from a plane for a set of points within a smaller local region at focus distance, D. ⁴	700 μm
Global Planarity Trueness	Average deviation from a plane in field of view at focus distance, D.	< 350 μm
Dimension Trueness	70-percentile dimension error in field of view at focus distance, D, and typical temperature range.	< 0.50 %
	70-percentile dimension error in field of view within optimal working distance and typical temperature range.	< 0.60 %
	70-percentile dimension error in field of view within optimal working distance and full temperature range.	< 0.70 %

⁴ Measured with Gaussian filter disabled.

FIGURE 15 – TYPICAL ZIVID ONE+ L POINT PRECISION VS. DISTANCE

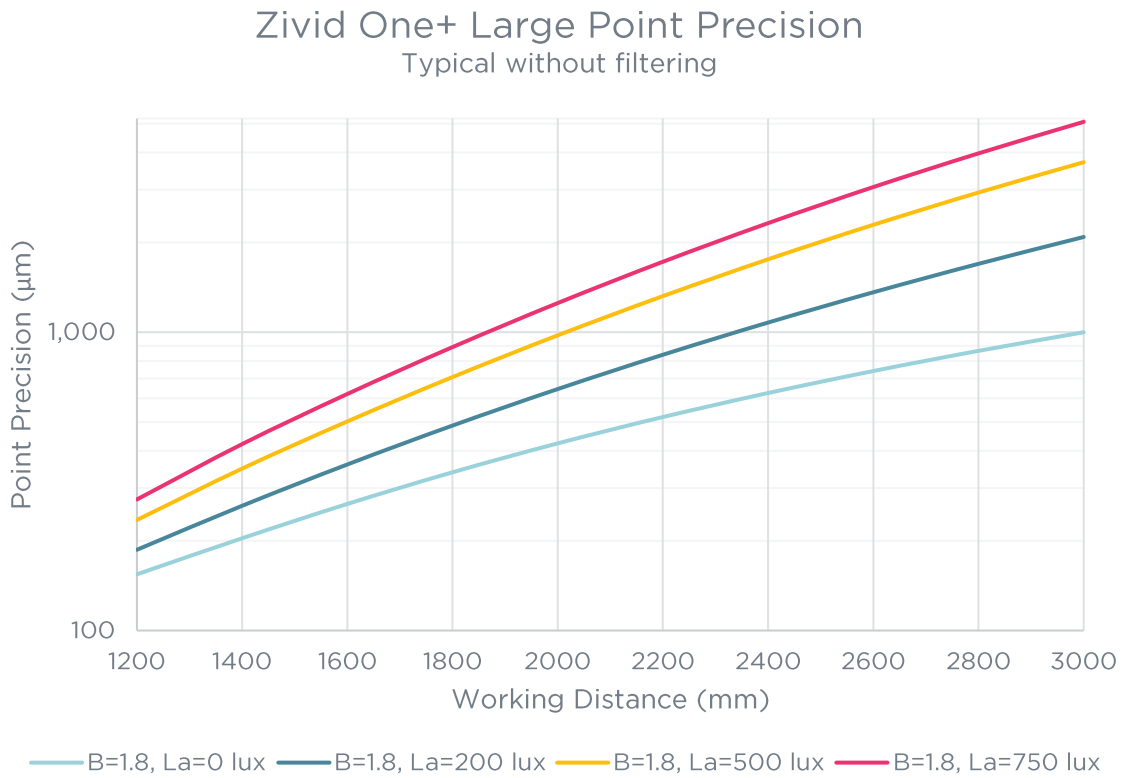


FIGURE 16 - TYPICAL ZIVID ONE+ L LOCAL PLANARITY PRECISION VS. DISTANCE

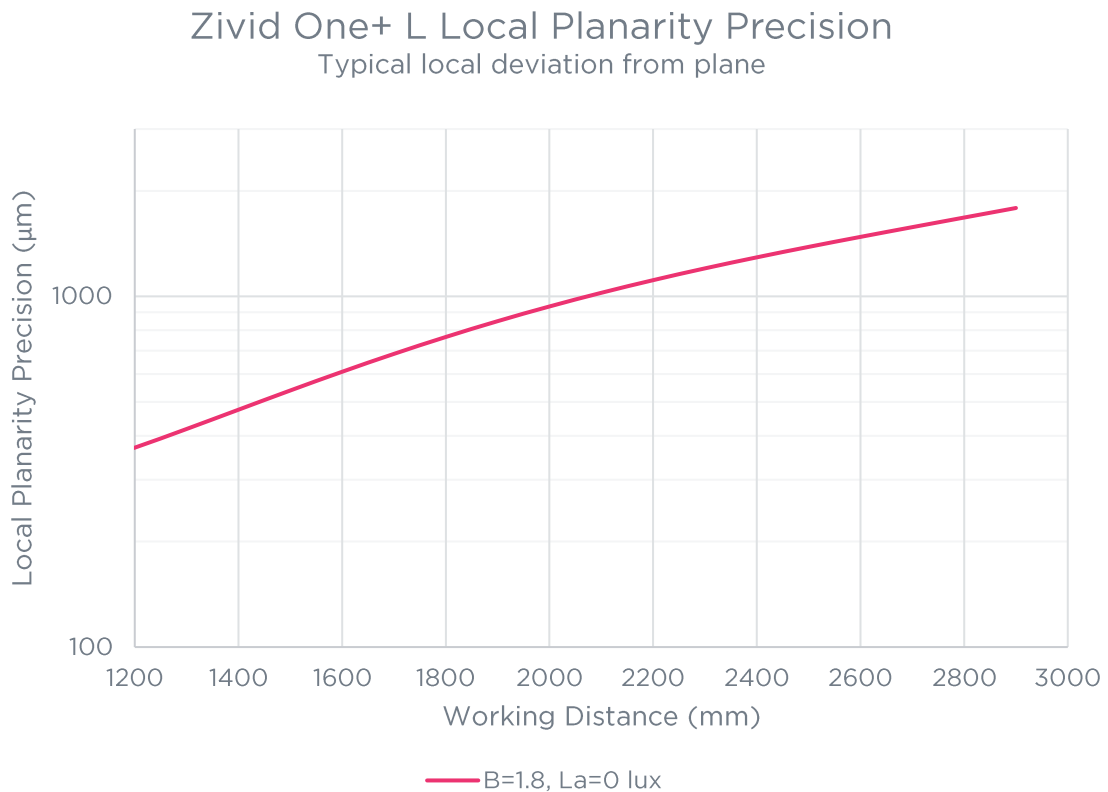


FIGURE 17 – TYPICAL ZIVID ONE+ L GLOBAL PLANARITY TRUENESS VS. DISTANCE

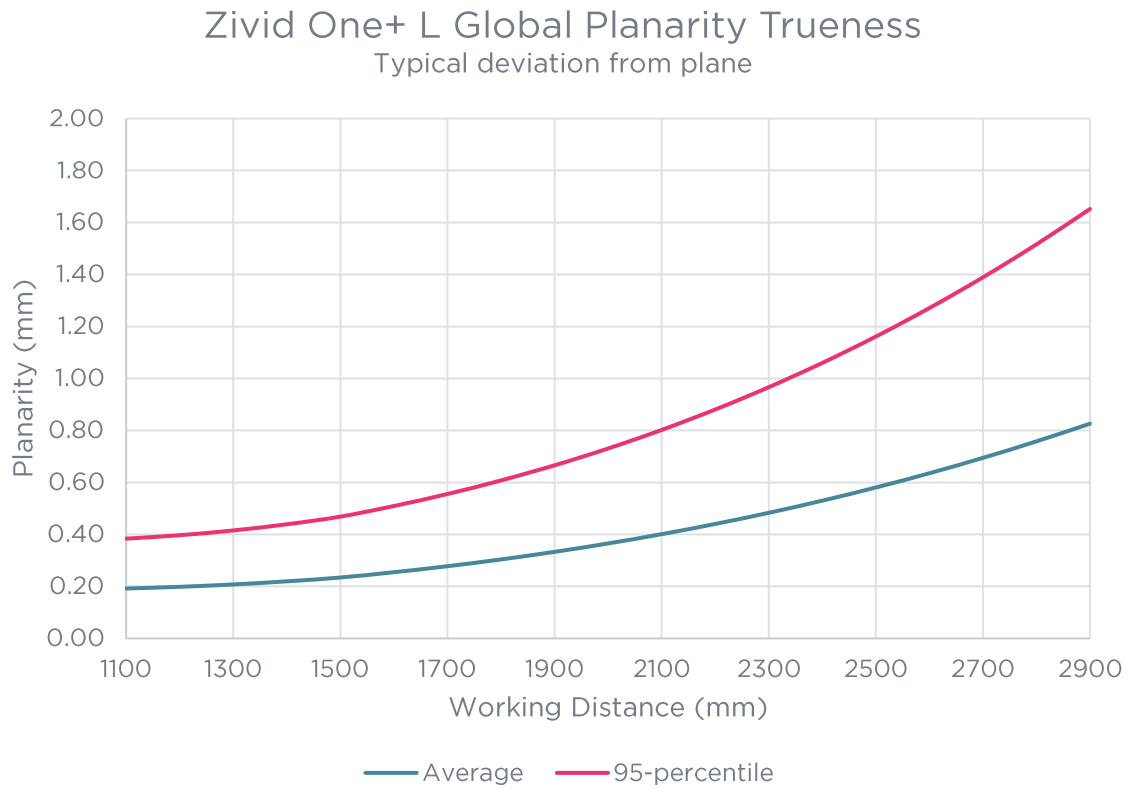
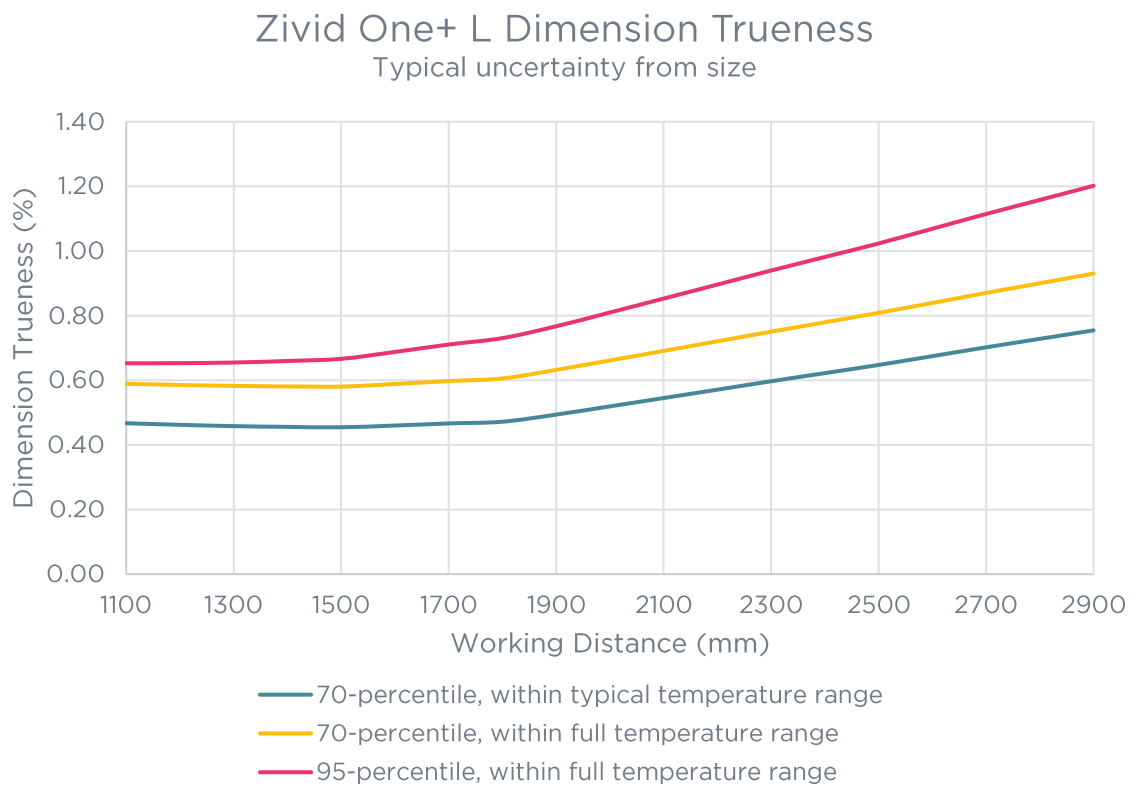


FIGURE 18 – TYPICAL ZIVID ONE+ L DIMENSION TRUENESS VS. DISTANCE



Physical specifications

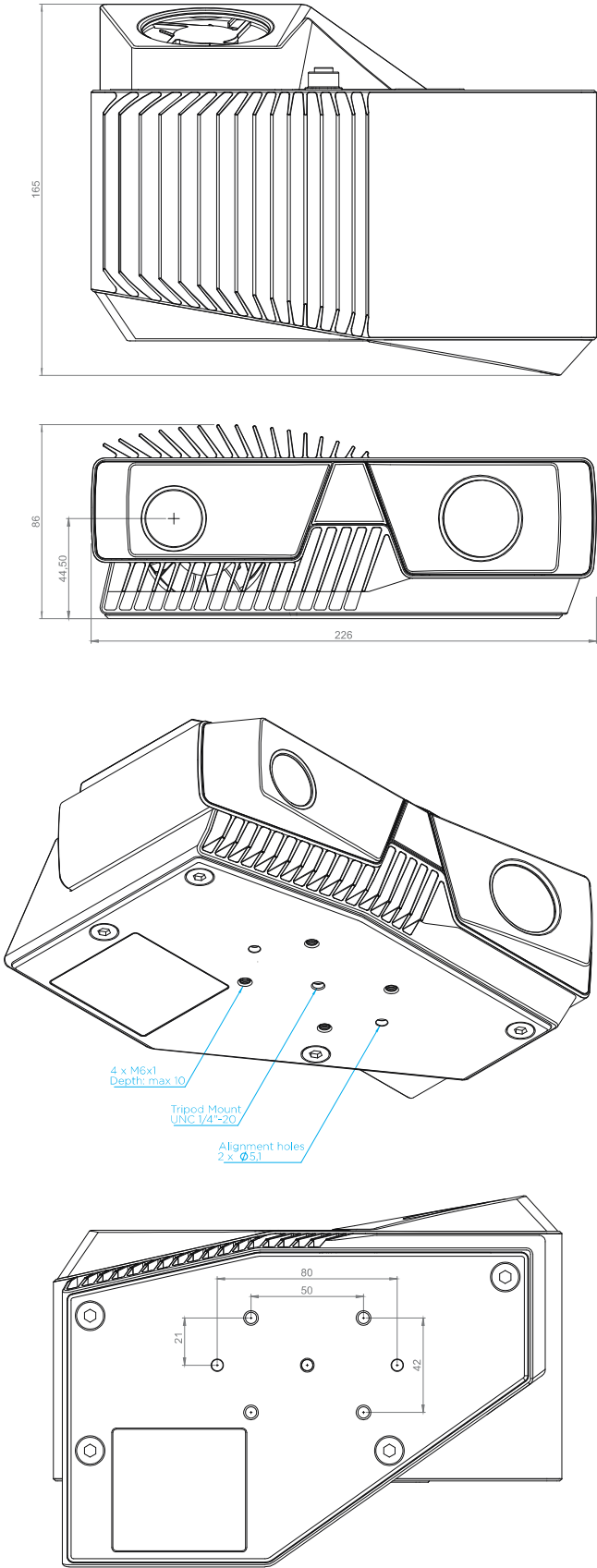
Size	226 mm x 165 mm x 86 mm
Weight	2 kg
Cable strain limit, power	90 N
Cable strain limit, data	30 N
Environmental	IP65 5 g sinusoidal ⁵ 15 g shock ⁶
Operating temperature	10° to 40° C
Storage temperature	-20° to 60° C
Data connector	USB 3.0 SuperSpeed USB Type B Jack screw M2
Power connector	M12-5
Power adapter	24V = 5A EU, US, and UK power plug options
Power consumption, typical	15 W, Idle 45 W, TDP ⁷ 120 W, Peak

⁵ IEC 60068-2-6, 10-150 Hz, 5 g, in X, Y and Z direction, 2 hour per axis. Sweep rate 1 octave per minute sweep rate.

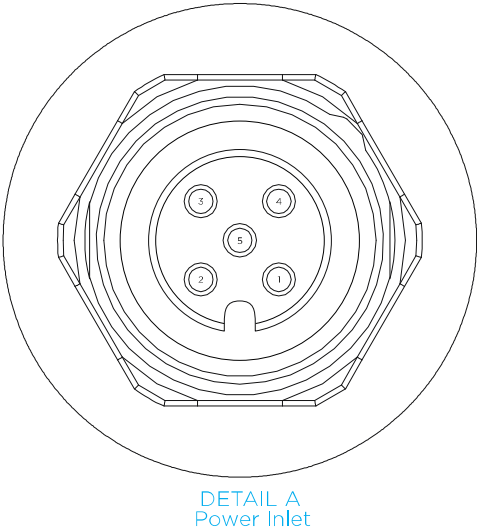
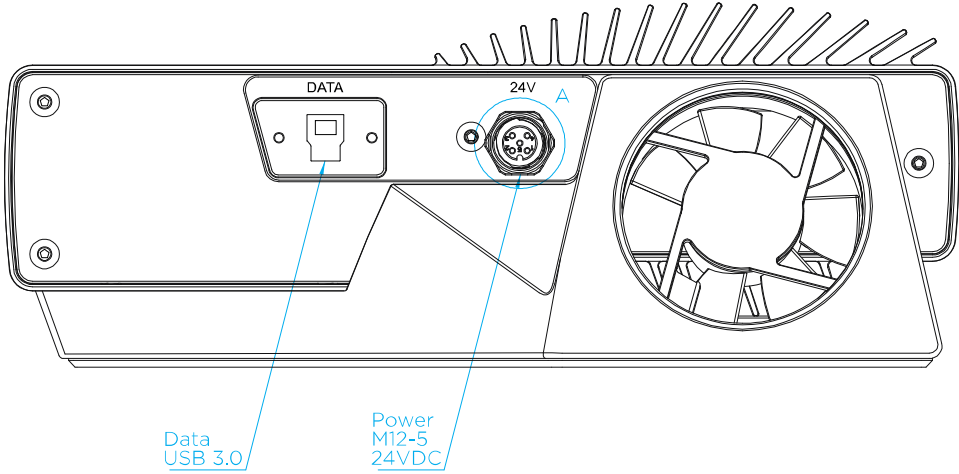
⁶ IEC 60068-2-27, 15 g / 11 ms half sine shock pulses. 3 shocks per direction, 18 shocks in total.

⁷ Thermal Design Power is the maximum power consumed by the camera when capturing 3D images in a continuous stream.

Mechanical drawings



Connectors



Pin

1	24V DC +/- 20% Max 4A
2	24V DC +/- 20% Max 4A
3	GND
4	GND
5	NC

Revision history

Ver.	Date	Notes
1.0	05/21	Updated front page Updated table of contents Updated "General specifications" and added valid revision number Updated "Operating distance and field of view" Added figure 2-16 Updated "Accuracy specifications" and table "Common conditions" Updated table "Zivid One+ S Typical Specifications" Updated table "Zivid One+ M Typical Specifications" Updated table "Zivid One+ L Typical Specifications" Updated table "Physical specifications"
0.91	05/19	Added metrics for Zivid One+ Small. Added metrics for Zivid One+ Large. Updated Operating distance and field of view table. Updated Zivid One+ Medium spec. plots. Updated connector table.
0.9	04/19	Initial version.

Zivid AS
Gjerdrums vei 10A
N0484 Oslo
Norway

© 2019 Zivid AS. All rights reserved. Subject to change without notice.

Appendix D

Move Group python interface test

```
#!/usr/bin/env python

import sys
import os
import copy
import rospy
import tf
import moveit_commander
import moveit_msgs.msg
import geometry_msgs.msg
from math import pi
from std_msgs.msg import String
from moveit_commander.conversions import pose_to_list
import numpy as np
import tf.transformations

#Initialize moveit_commander
def main():
    im_num=0
    moveit_commander.roscpp_initialize(sys.argv)
    rospy.init_node('robot_irb', anonymous=True)

robot = moveit_commander.RobotCommander()

#Instantiate a PlanningSceneInterface object. This provides a remote ...
    interface for getting,
scene = moveit_commander.PlanningSceneInterface()

#define the groupname
group_name = 'righty_tcp'
move_group = moveit_commander.MoveGroupCommander(group_name)
move_group.set_planner_id('RRT')
#move_group.set_planning_time(10)
#move_group.set_goal_position_tolerance(0.001)
#move_group.set_goal_orientation_tolerance(0.005)

#define the Publisher
display_trajectory_publisher = ...
    rospy.Publisher('/move_group/display_planned_path', ...
    moveit_msgs.msg.DisplayTrajectory, queue_size=20)

print ("Taking first capture")
#print (move_group.get_current_pose())
#print (move_group.get_current_rpy())
```

```

pose_goal = geometry_msgs.msg.Pose()
#pose_goal = move_group.get_random_pose()
#First position
pose_goal.orientation.x = 0.353
pose_goal.orientation.y = 0.612
pose_goal.orientation.z = 0.353
pose_goal.orientation.w = 0.612
pose_goal.position.x = 3.76 #3.8
pose_goal.position.y = 5.0
pose_goal.position.z = 0.83

#Move robot to the first position
move_group.set_pose_target(pose_goal)

plan = move_group.go(wait=True)

# Calling `stop()` ensures that there is no residual movement
move_group.stop()
rospy.sleep(2)

move_group.clear_pose_targets()

#Second position
#q = tf.transformations.quaternion_from_euler(0,1.5707963, 0)
pose_goal.orientation.x = 0#q[0]
pose_goal.orientation.y = 0.707#q[1]
pose_goal.orientation.z = 0#q[2]
pose_goal.orientation.w = 0.707#q[3]
pose_goal.position.x = 3.76
pose_goal.position.y = 5.0
pose_goal.position.z = 0.83 #0.74

#Move robot to the third position
move_group.set_pose_target(pose_goal)
plan = move_group.go(wait=True)

# Calling `stop()` ensures that there is no residual movement
move_group.stop()
rospy.sleep(2)

move_group.clear_pose_targets()

if __name__ == '__main__':
    main()

```

Appendix E

Software workflow

Lenovo Legion 5 laptop RTX3070 Software setup

Graphics driver

“Activities -> Software & Update -> Additional drivers”

Installed nvidia 460 driver -> Blackscreen

Installed nvidia 470 driver -> Success

Must be done in order to utilize GPU computing.

Zivid

Installed Zivid core, tools, SDK etc. from <https://github.com/zivid/zivid-ros>. Read the guide with caution. Tested camera from GitHub guide, worked fine.

MoveIt

Installed MoveIt, and MoveIt visual tools

Installed industrial core, and abb driver

Copied configuration files from 16.04 computer: Launch file works.

Robot Config

Connected with robot. Adress: 10.225.120.54 (righty). Further connection setup explained in the report.

This worked for us, and the robot could then be controlled from a laptop.

YOLOv5 requirements

Installed all necessary python packages from requirements.txt from <https://github.com/ultralytics/yolov5>. However, Pytorch/torchvision was installed separately. See further down. Most important notice is that YOLOv5 require python 3. ROS on the other hand operates with python 2.

CUDA

- Failed first try and had two graphics drivers installed = blackscreen. Fresh install of ubuntu, and started from scratch.
- CUDA installation

The CUDA version to be installed depends on the graphics driver in use (<https://docs.nvidia.com/deploy/cuda-compatibility/>)

With the nvidia 470 driver in use (RTX 3070), CUDA 11.4 (<https://developer.nvidia.com/cuda-toolkit-archive>) was installed.

The easiest way to do this is with the runfile option:

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System: Linux, Windows

Architecture: x86_64, ppc64le, sbsa

Distribution: CentOS, Debian, Fedora, OpenSUSE, RHEL, SLES, Ubuntu, WSL-Ubuntu, conda

Version: 18.04, 20.04

Installer Type: deb (local), deb (network), runfile (local)

Download Installer for Linux Ubuntu 18.04 x86_64

The base installer is available for download below.

Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/11.4.1/local_installers/cuda_11.4.1_470.57.02_linux.run
$ sudo sh cuda_11.4.1_470.57.02_linux.run
```

After the installation, follow section 3 and 4 in this guide: <https://medium.com/@anarmammadli/how-to-install-cuda-11-4-on-ubuntu-18-04-or-20-04-63f3dee2099>

Package fixes

- Did the previous steps before CUDA due to the fresh install.
- When installing python3 (<http://dhanoopbhaskar.com/blog/2020-05-07-working-with-python-3-in-ros-kinetic-or-melodic/>), some packages will be removed and we have to reinstall them
 - sudo apt install ros-melodic-desktop-full --fix-missing
 - sudo apt install ros-melodic-moveit --fix-missing
 - sudo apt install ros-melodic-abb --fix-missing
 - sudo apt install ros-melodic-industrial-core --fix-missing

When executing python3 scripts, include this shebang: `#!/usr/bin/env python3`

PyTorch

- Pytorch
 - Yolov5 utilize pytorch. Your pytorch version must be compatible with the CUDA version you are running. However, we got away with downloading Pytorch 1.10.2. (<https://pytorch.org/>)

PyTorch Build	Stable (1.10.2)	Preview (Nightly)	LTS (1.8.2)
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python	C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.3	ROCm 4.2 (beta)
Run this Command:	<pre>pip3 install torch==1.10.2+cu113 torchvision==0.11.3+cu113 torchaudio==0.10.2+cu113 -f https://download.pytorch.org/whl/cu113/torch_stable.html</pre>		

- In order to check if your GPU is used by torch:
<https://stackoverflow.com/questions/48152674/how-to-check-if-pytorch-is-using-the-gpu>

```

josteink@josteink: ~
File Edit View Search Terminal Help
josteink@josteink:~$ python3
Python 3.6.9 (default, Dec 8 2021, 21:08:43)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.current_device()
0
>>> torch.cuda.device(0)
<torch.cuda.device object at 0x7f811968a208>
>>> torch.cuda.device_count()
1
>>> torch.cuda.get_device_name(0)
'NVIDIA GeForce RTX 3070 Laptop GPU'
>>>

```

CuDNN

The compatible cuDNN version for CUDA 11.4 is v8.4.0. A Nvidia account must be made, if you do not have one.

cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

[Download cuDNN v8.4.0 \(April 1st, 2022\), for CUDA 11.x](#)

[Download cuDNN v8.4.0 \(April 1st, 2022\), for CUDA 10.2](#)

[Archived cuDNN Releases](#)

Bibliography

- [1] A. Koprivc testXpo 2019. *Mechanical Testing Solutions for Lithium-Ion batteries in Automotive applications*. URL: https://www.zwickroell.com/fileadmin/content/Files/PDF/testXpo/Vortraege/2019/ZR_Aleksander_Koprivc_eng.pdf. (accessed: 27.01.2022).
- [2] ABB. *IRC5 Industrial Robot Controller, data sheet, PDF*. URL: <https://new.abb.com/products/robotics/controllers/irc5>. (accessed: 24.05.2022).
- [3] ABB. *Technical data*. URL: <https://new.abb.com/products/robotics/industrial-robots/irb-4400>. (accessed: 24.05.2022).
- [4] Jameel et al. *The Effect of Laser Welding on the Tensile Strength and Radiographic Analysis of Co-Cr Repaired Joints*. URL: https://www.researchgate.net/publication/333163078_The_Effect_of_Laser_Welding_on_the_Tensile_Strength_and_Radiographic_Analysis_of_Co-Cr_Repaired_Joints. (accessed: 08.02.2022).
- [5] Thompson et al. "The importance of design in lithium ion battery recycling – a critical review." In: *Green Chem.* 22 (22 2020), pp. 7585–7603. DOI: 10.1039/D0GC02745F. URL: <http://dx.doi.org/10.1039/D0GC02745F>.
- [6] Harper G. et al. "Recycling lithium-ion batteries from electric vehicles." In: *Nature* 575.1 (2019), pp. 75–86. DOI: <https://doi.org/10.1038/s41586-019-1682-5>.
- [7] Youzi Xiao et. al. "A review of object detection based on deep learning." In: *Multimed Tools Appl* 79 (2020), pp. 23729–23791. DOI: <https://doi.org/10.1007/s11042-020-08976-6>.
- [8] Lihi Gur Arie. *The practical guide for Object Detection with YOLOv5 algorithm*. URL: <https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843>. (accessed: 10.05.2022).
- [9] Sabri Baazouzi et al. "Optimization of Disassembly Strategies for Electric Vehicle Batteries." In: *Batteries* 7.4 (2021). ISSN: 2313-0105. DOI: 10.3390/batteries7040074. URL: <https://www.mdpi.com/2313-0105/7/4/74>.
- [10] Second life ev batteries. *VW e-Golf 1.6kWh*. URL: <https://www.secondlife-evbatteries.com/collections/ev-battery/products/vw-egolf-15v-samsung-sdi>. (accessed: 01.02.2022).
- [11] Eduard Marti Bigorra. "Design and implementation of a computer vision system for robotic disassembly of electric vehicle battery pack." URL: https://uiano.sharepoint.com/:f:/r/teams/AULIBDIS/Delte%20dokumenter/Projects/Reports_2020?csf=1&web=1&e=na7B3e. (accessed: 10.04.2022).
- [12] Brainvoyager. *Spatial Transformation Matrices*. URL: <https://www.brainvoyager.com/bv/doc/UsersGuide/CoordsAndTransforms/SpatialTransformationMatrices.html>. (accessed: 01.05.2022).
- [13] Jason Brownlee. *Overfitting and Underfitting With Machine Learning Algorithms*. URL: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>. (accessed: 06.05.2022).
- [14] Jason Brownlee. *What is the difference between a Batch and an Epoch in a Neural Network*. URL: https://deeplearning.lipingyang.org/wp-content/uploads/2018/07/What-is-the-Difference-Between-a-Batch-and-an-Epoch-in-a-Neural-Network_.pdf. (accessed: 06.05.2022).

- [15] Boston Consulting Group BSG. *Robotics Outlook 2030*. URL: <https://www.bcg.com/publications/2021/how-intelligence-and-mobility-will-shape-the-future-of-the-robotics-industry>. (accessed: 30.04.2022).
- [16] Rishav Chakravarti and Xiannong Meng. “A Study of Color Histogram Based Image Retrieval.” In: *2009 Sixth International Conference on Information Technology: New Generations*. 2009, pp. 1323–1328. DOI: [10.1109/ITNG.2009.126](https://doi.org/10.1109/ITNG.2009.126).
- [17] Martin Choux, Eduard Marti Bigorra, and Ilya Tyapin. “Task Planner for Robotic Disassembly of Electric Vehicle Battery Pack.” In: *Metals* 11.3 (2021). ISSN: 2075-4701. DOI: [10.3390/met11030387](https://doi.org/10.3390/met11030387). URL: <https://www.mdpi.com/2075-4701/11/3/387>.
- [18] Atle Christiansen. *Universitetet i Agder demonterer el-bilbatterier for gjenbruk*. URL: <https://www.uia.no/nyheter/universitetet-i-agder-demonterer-el-bilbatterier-for-gjenbruk>. (accessed: 11.01.2022).
- [19] David Coleman et al. “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study.” In: *CoRR* abs/1404.3785 (2014). arXiv: [1404.3785](https://arxiv.org/abs/1404.3785). URL: <http://arxiv.org/abs/1404.3785>.
- [20] Teng Deng et al. “Multiple consumer-grade depth camera registration using everyday objects.” In: *Image and Vision Computing* 62 (2017), pp. 1–7. ISSN: 0262-8856. DOI: <https://doi.org/10.1016/j.imavis.2017.03.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0262885617300677>.
- [21] Nvidia Developer. *Nvidia cuDNN*. URL: <https://developer.nvidia.com/cudnn>. (accessed: 13.05.2022).
- [22] Anamika Dhillon and Gyanendra K. Verma. “Convolutional neural network: a review of models, methodologies and applications to object detection.” In: *Prog Artif Intell* 9 (2020), pp. 85–112. DOI: <https://doi.org/10.1007/s13748-019-00203-0>.
- [23] Allen Elizabeth and Triantaphillidou Sophie. *The Manual of Photography*. Vol. 10th ed. Routledge, 2011. ISBN: 9780240520377. URL: <https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=505943&site=ehost-live>.
- [24] Grasp Engineering. *What is linear static analysis*. URL: <https://www.graspengineering.com/what-is-linear-static-analysis/>. (accessed: 01.05.2022).
- [25] Eramet. *AN INNOVATIVE PROCESS FOR RECYCLING LITHIUM-ION BATTERIES FROM ELECTRIC VEHICLES*. URL: <https://www.eramet.com/en/activities/innovate-design/relieve-project>. (accessed: 03.02.2022).
- [26] Evans, Jonathan Gwynne Blackledge, and Yardley Peter. “Numerical Methods for Partial Differential Equations.” In: *Prog Artif Intell* 9 (2000), pp. 85–112. DOI: <https://doi.org/10.1007/978-1-4471-0377-6..>
- [27] Florian Fournier and Jannick Rolland. “Design Methodology for High Brightness Projectors.” In: *Journal of Display Technology* 4.1 (2008), pp. 86–91. DOI: [10.1109/JDT.2007.907110](https://doi.org/10.1109/JDT.2007.907110).
- [28] Silvio Giancola. *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. eng. Cham, 2018.
- [29] Yang Hua et al. “Toward Sustainable Reuse of Retired Lithium-ion Batteries from Electric Vehicles.” In: *Resources, Conservation and Recycling* 168 (2021), p. 105249. ISSN: 0921-3449. DOI: <https://doi.org/10.1016/j.resconrec.2020.105249>. URL: <https://www.sciencedirect.com/science/article/pii/S0921344920305644>.
- [30] Hydrovolt.
- [31] Kweon Hyeong. “Determination of true stress-strain curve of type 304 and 316 stainless steels using a typical tensile test and finite element analysis.” In: *Nuclear Engineering and Technology*. 53. [10.1016/j.net.2020.07.014](https://doi.org/10.1016/j.net.2020.07.014). (). URL: https://www.researchgate.net/publication/343309612_Determination_of_true_stress-strain_curve_of_type_304_and_316_stainless_steels_using_a_typical_tensile_test_and_finite_element_analysis. (accessed: 14.05.2022).

- [32] ROS Industrial. *abb_robot_driver*. URL: https://github.com/ros-industrial/abb_robot_driver. (accessed: 23.05.2022).
- [33] JournalDev. *What is .bashrc file in Linux?* URL: <https://www.journaldev.com/41479/bashrc-file-in-linux>. (accessed: 12.04.2022).
- [34] Margrit Kasper-Eulaers et al. "Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5." In: *Algorithms* 14.4 (2021). ISSN: 1999-4893. DOI: [10.3390/a14040114](https://doi.org/10.3390/a14040114). URL: <https://www.mdpi.com/1999-4893/14/4/114>.
- [35] Paulo Roberto Gardel Kurka and Aldo André Díaz Salazar. "Applications of image processing in robotics and instrumentation." In: *Mechanical Systems and Signal Processing* 124 (2019), pp. 142–169. ISSN: 0888-3270. DOI: <https://doi.org/10.1016/j.ymsp.2019.01.015>. URL: <https://www.sciencedirect.com/science/article/pii/S0888327019300160>.
- [36] Mohammad Ali Mahdavi-pour. "Failure prediction of steel components under the coupled effect of excessive plastic deformations and pitting corrosion." In: (2022).
- [37] P et. al. Moazzam. "Lithium bioleaching: An emerging approach for the recovery of Li from spent lithium ion batteries." In: *Chemosphere* 277.1 (2021), pp. 130–196. DOI: <https://doi.org/10.1016/j.chemosphere.2021.130196>.
- [38] MoveIt. *Move Group Python Interface*. URL: http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/move_group_python_interface/move_group_python_interface_tutorial.html. (accessed: 19.04.2022).
- [39] MoveIt! *Concepts*. URL: <https://moveit.ros.org/documentation/concepts/>. (accessed: 26.05.2022).
- [40] Oyyaravelu et al. N. "Metallurgical and mechanical properties of laser welded high strength low alloy steel." In: *Journal of advanced research* 7.3 (2016), pp. 463–472. DOI: <https://doi.org/10.1016/j.jare.2016.03.005>.
- [41] N/A. *Quaternions*. URL: <https://quaternions.online/>. (accessed: 19.04.2022).
- [42] Joseph Nelson. *YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS*. URL: <https://blog.roboflow.com/yolov5-is-here/>. (accessed: 13.05.2022).
- [43] Joseph Nelson. *Your Comprehensive Guide to the YOLO Family of Models*. URL: <https://blog.roboflow.com/guide-to-yolo-models/>. (accessed: 11.05.2022).
- [44] Thanh Chi Nguyen. "Automated Li-Ion Battery Dismantling Line (From Module To Cells)." URL: https://uiano.sharepoint.com/:f:/r/teams/AULIBDIS/Delte%20dokumenter/Projects/Past_projects/Report_2018?csf=1&web=1&e=PV4PV8. (accessed: 02.02.2022).
- [45] Jan Peder Nilsen. "Computer vision system to remove EV battery lid using an industrial robot." URL: https://uiano.sharepoint.com/:f:/r/teams/AULIBDIS/Delte%20dokumenter/Projects/Reports_2020?csf=1&web=1&e=na7B3e. (accessed: 02.05.2022).
- [46] Carol L Novak, Steven A Shafer, et al. "Anatomy of a color histogram." In: *CVPR*. Vol. 92. 1992, pp. 599–605.
- [47] Nvidia. *PyTorch*. URL: <https://www.nvidia.com/en-us/glossary/data-science/pytorch/>. (accessed: 13.05.2022).
- [48] Developer Nvidia. *Your GPU Compute Capability*. URL: <https://developer.nvidia.com/cuda-gpus>. (accessed: 16.05.2022).
- [49] Developer Zone Nvidia. *CUDA Compatibility*. URL: <https://docs.nvidia.com/deploy/cuda-compatibility/index.html>. (accessed: 17.05.2022).
- [50] OFV. *2021 ble tidenes rekordår for nybilregistreringer*. URL: <https://ofv.no/>. (accessed: 11.01.2022).
- [51] Fred Oh. *What Is CUDA*. URL: <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/>. (accessed: 13.05.2022).

- [52] Zivid One+. *Datasheet: Technical specifications*. URL: <https://www.zivid.com/zivid-one-plus>. (accessed: 24.05.2022).
- [53] Arthur Ouaknine. *Review of Deep Learning Algorithms for Object Detection*. URL: <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>. (accessed: 26.05.2022).
- [54] Eli Peli. "Contrast in complex images." In: *J. Opt. Soc. Am. A* 7.10 (Oct. 1990), pp. 2032–2040. DOI: [10.1364/JOSAA.7.002032](https://doi.org/10.1364/JOSAA.7.002032). URL: <http://opg.optica.org/josaa/abstract.cfm?URI=josaa-7-10-2032>.
- [55] ArcGIS Pro. *How the Compute Accuracy For Object Detection tool works*. URL: <https://pro.arcgis.com/en/pro-app/2.8/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm>. (accessed: 14.05.2022).
- [56] Morgan Quigley. "ROS: an open-source Robot Operating System." In: *ICRA 2009*. 2009.
- [57] *Recycling of Spent Lithium-Ion Batteries : Processing Methods and Environmental Impacts*. eng. Cham, 2019.
- [58] RoboDK. *Basic Guide*. URL: <https://robodk.com/doc/en/Basic-Guide.html>. (accessed: 08.02.2022).
- [59] ROS.org. *Distributions*. URL: <http://wiki.ros.org/Distributions>. (accessed: 24.05.2022).
- [60] ROS.org. *Move Group Python Interface*. URL: http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/move_group_python_interface/move_group_python_interface_tutorial.html. (accessed: 26.05.2022).
- [61] ROS.org. *MoveIt Setup Assistant*. URL: http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/setup_assistant/setup_assistant_tutorial.html#overview. (accessed: 26.05.2022).
- [62] ROS.org. *ROS installation options*. URL: <http://wiki.ros.org/ROS/Installation>. (accessed: 10.04.2022).
- [63] ROS.org. *Saving images with image_saverwithtimestamp*. URL: https://answers.ros.org/question/283724/saving-images-with-image_saver-with-timestamp/. (accessed: 02.05.2022).
- [64] ROS.org. *URDF and SRDF*. URL: http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/urdf_srdf/urdf_srdf_tutorial.html. (accessed: 26.05.2022).
- [65] Rufflewind's Scratchpad. *Reverse-mode automatic differentiation: a tutorial*. URL: <https://rufflewind.com/2016-12-30/reverse-mode-automatic-differentiation>. (accessed: 13.05.2022).
- [66] Jacob Solawetz. *How to Train A Custom Object Detection Model with YOLO v5*. URL: <https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5-917e9ce13208>. (accessed: 10.05.2022).
- [67] Jacob Solawetz. *Train, Validation, Test Split for Machine Learning*. URL: <https://blog.roboflow.com/train-test-split/>. (accessed: 14.05.2022).
- [68] Chamidu Supeshala. *YOLO v4 or YOLO v5 or PP-YOLO?* URL: <https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109>. (accessed: 11.05.2022).
- [69] László Szécsi, Mateu Sbert, and László Szirmay-Kalos. "Combined Correlated and Importance Sampling in Direct Light Source Computation and Environment Mapping." In: *Computer Graphics Forum* 23.3 (2004), pp. 585–593. DOI: <https://doi.org/10.1111/j.1467-8659.2004.00790.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2004.00790.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2004.00790.x>.

- [70] Wei Jie Tan et al. “A hybrid disassembly framework for disassembly of electric vehicle batteries.” In: *International Journal of Energy Research* 45.5 (2021), pp. 8073–8082. DOI: <https://doi.org/10.1002/er.6364>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/er.6364>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/er.6364>.
- [71] Ilya Tyapin, Knut Berg Kaldestad, and Geir Hovland. “Off-line path correction of robotic face milling using static tool force and robot stiffness.” In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 5506–5511. DOI: [10.1109/IROS.2015.7354157](https://doi.org/10.1109/IROS.2015.7354157).
- [72] Ultralytics. *yolov5*. URL: <https://github.com/ultralytics/yolov5>. (accessed: 06.05.2022).
- [73] Ultralytics. *YOLOv5 Documentation*. URL: <https://docs.ultralytics.com/>. (accessed: 11.05.2022).
- [74] YoloV5 Ultralytics. *Tips for Best Training Results*. URL: <https://github.com/ultralytics/yolov5/wiki/Tips-for-Best-Training-Results>. (accessed: 06.05.2022).
- [75] YoloV5 Ultralytics. *Train Custom Data*. URL: <https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data>. (accessed: 17.05.2022).
- [76] Diego Unzueta. *Convolutional Layers vs Fully Connected Layers*. URL: <https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b>. (accessed: 26.05.2022).
- [77] Supachai Vongbunyong and Wei Hua Chen. “General Disassembly Process.” In: *Disassembly Automation: Automated Systems with Cognitive Abilities*. Cham: Springer International Publishing, 2015, pp. 9–24. ISBN: 978-3-319-15183-0. DOI: [10.1007/978-3-319-15183-0_2](https://doi.org/10.1007/978-3-319-15183-0_2). URL: https://doi.org/10.1007/978-3-319-15183-0_2.
- [78] Material Welding. *What is Heat Affected Zone or HAZ in Welding*. URL: <https://www.materialwelding.com/what-is-heat-affected-zone-or-haz-in-welding-haz-width/>. (accessed: 15.02.2022).
- [79] Li Yong-Jia, Lei Ting, and Zeng Gui-sheng. “Hydrometallurgical process for recovery and synthesis of LiCoO₂ from spent lithium-ion batteries.” In: *2011 International Conference on Electric Technology and Civil Engineering (ICETCE)*. 2011, pp. 6009–6011. DOI: [10.1109/ICETCE.2011.5775378](https://doi.org/10.1109/ICETCE.2011.5775378).
- [80] Pietro Zanuttigh. *Time-of-Flight and Structured Light Depth Cameras : Technology and Applications*. eng. Cham, 2016.
- [81] Hui Zhang et al. “Machining with flexible manipulator: toward improving robotic machining performance.” In: 3 (2005). DOI: [10.1109/AIM.2005.1511161](https://doi.org/10.1109/AIM.2005.1511161). URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1511161&tag=1>.
- [82] Zhong-Qiu Zhao et al. “Object Detection With Deep Learning: A Review.” In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11 (2019), pp. 3212–3232. DOI: [10.1109/TNNLS.2018.2876865](https://doi.org/10.1109/TNNLS.2018.2876865).
- [83] Zivid. *Zivid One+ Medium*. URL: <https://www.zivid.com/zivid-one-plus-medium-3d-camera>. (accessed: 02.05.2022).
- [84] Zivid. *Zivid ROS driver*. URL: <https://github.com/zivid/zivid-ros>. (accessed: 26.04.2022).
- [85] Support Zivid. *GPU requirements*. URL: <https://support.zivid.com/latest/getting-started/software-installation/gpu/gpu-requirements.html>. (accessed: 16.05.2022).
- [86] Zivid Knowledge Base Zivid. *Acquisition Settings*. URL: <https://support.zivid.com/latest/reference-articles/settings/acquisition-settings.html>. (accessed: 29.04.2022).