Czech
Technical
University
in Prague

Department of Cybernetics

Knowledge-based approach for robotic assembly of printed circuits

Vojtěch Janů

Supervisor: Petr Kadera, Ph.D
May 2018

Acknowledgements

I would like to thak to my supervisor for his guidance, my family and friends for their support and my alma mater CTU for giving me this oportunity.

Declaration

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, May 25, 2018

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 25. května 2018

Abstract

This thesis discusses the knowledge-based approach for robotic assembly of electronic circuits. For this purpose Architecture with use of ontology as a source of information with skill representation of robotic unit, possibilities are considered. In this thesis solution to the problem of product assembly that is based on PDDL planner is purposed. Thus, no exact sequence of operations is needed in the product description. For the purpose of robot manipulation, calibration methods are suggested and implemented. For the approach verification supportive tools such as fingers for gripper or circuit board holders are designed and tested with use of Finite Element Analysis. Purposed architecture has been implemented and experimentally verified.

Keywords: ontology, KUKA iiwa, PDDL, planner, industry 4.0, 3D print, PCB

Supervisor: Petr Kadera, Ph.D

Abstrakt

Tato práce se zabývá přístupem založeným na znalostech pro osazování desek plošných spojů. K tomuto účelu je zvážena architektura s ontologií sloužící jako zdroj informací a s reprezentací dovedností pro robotickou buňku. V této práci je navrženo řešení založené na PDDL plánovači. Díky plánovači nemusí být přesné pořadí operací zaneseno přímo v popisu produktu, ale může být generováno s ohledem na možnosti robotické jednotky. Pro potřeby robotické manipulace byly navrženy a vyzkoušeny postupy pro kalibraci. Pro potřeby ověření zvoleného přístupu byly navrženy a otestovány pomocné nástroje, jako například prsty chapadla, nebo držáky desek plošných spojů. Tyto nástroje byly otestovány pomocí analýzy konečného počtu elementů. Navržená architektura byla implementována a experimentálně ověřena.

Klíčová slova: ontologie, KUKA iiwa, PDDL, plánovač, průmysl 4.0, 3D tisk, desky plošných spojů

Překlad názvu: Přístup založený na znalostech pro robotické osazování plošných spojů

Contents

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Janů** Jméno: **Vojtěch** Osobní číslo: **406464**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra řídicí techniky**

Studijní program: **Kybernetika a robotika**

Studijní obor: **Kybernetika a robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Přístup založený na znalostech pro robotické osazování plošných spojů

Název diplomové práce anglicky:

Knowledge-based approach for robotic assembly of printed circuits

Pokyny pro vypracování:

Seznam doporučené literatury:

[1] Baader Franz et al. The Description Logic Handbook - Cambridge, 2007
[2] Brachman Ronald, Levesque Hector - Knowledge Representation and Reasoning, San Francisco, 2004
[3] Bruno Siciliano, Oussama Khatib - Springer Handbook of Robotics - Springer-Verlag Berlin Heidelberg, 2008
[4] Sebastian Thrun, Wolfram Burgard, Dieter Fox - Probabilistic Robotics - The MIT Press, 2005

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Petr Kadera, Ph.D., inteligentní systémy pro průmysl a chytré distribuční sítě CIIRC

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **16.01.2018** Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **30.09.2019**

Ing. Petr Kadera, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Ing. Pavel Ripka, CSc.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání Podpis studenta

Chapter 1

Introduction

Since the third industrial revolution, many things have changed. Technology has envolved and with that new needs for this century have arisen. Nowadays companies are standing in front of lack of workforce [Job],[Man], the need for product customization and flexible manufacturing, the need for acceleration of launch cycle for the new product. Also, especially for small and middle size companies, there is a need for inexpensive new technology deployment[BFKR14], [WSOG15]. All these problems challenge researchers all around the globe to develop new technologies, new processes and new software to overlap this gap on the market.

Recently, a new generation of industrial robots has entered the market. In the past industrial robots had to work in a work cell separately from human workers. These cells were needed because of the lack of feedback information about forces applied to robot arm, therefore robot could harm a human worker. This new generation of the industrial robots allows workers to work side by side with their robot companions. Those robots are called collaborative robots or cobots. Thanks to mechanically driven feedback, for example in case of Festo BionicCobot[FES], or sensor driven feedback, in case of Kuka iiwa[Gmbb] or ABB YuMi [Rob], collaborative robots stop or flex in case of collision and are therefore unharmful to human coworkers. These robots are designed to do repetitive and precise tasks, so human workers can utilize their complex perception, high dexterity, and common sense when working at one place. Thanks to this cooperation, companies can save much money for what would cost fully automated flexible solution or a lot of time and accuracy for what would cost solution without robot at all.

Replacing noncollaborative robot with collaborative one adds some flexibility to the manufacturing process, but it is not enough. In some cases, use of collaborative robot does not make sense and even if it does there is a problem. If the manufacturing process changes, someone must reprogram the robot. Currently, industrial robots are programmed by experts to follow specific motions in a highly structured environment and every reprograming is very expensive. That is the reason why many researchers are trying to simplify the process of reconfiguration and programming of robotic units. One approach to the simplification is to develop "skills" [MNB13], [Ste11], [Jac15] that covers all possibilities of the robotic unit and from which any task that needs to be done on that unit has to be composed. In some approaches[PHK14] accomplishment of every elementary skill needs to be evaluable and in others[Ste15] the evaluation or inspection is taken as the skill itself. The use of skills allows to a programmer to work on a higher level, only with skills. At the same time production line can be used for more products or more modifications of products at once just by definition of the sequence of skills needed for each product. All of this can lead to the process, where the whole code is generated autonomously based on detailed enough product description made from unique customer order.

1.1 Motivation

The end result that we would like to come to is a scenario where the description of the product would be all that is needed for the factory to decide if it can make the desired outcome or not. If the factory is able to do it, it would plan an effective process of manufacturing and manage all of its resources to produce the desired product. In this chain, there is the need for the description of the product to be descriptive enough to provide information about requirements of a process that needs to be done to produce the desired product. At the same time, all tools, skills, and processes in the factory need to be described in a way that reasoning unit can try to find a valid sequence of commands to fulfill those requirements. For this purpose, ontology is a good candidate to describe all relations and data about the product as well as about possibilities of the factory. Because there are the DL ontologies the reasoner can be run over and decide about fulfilling requirements for manufacturing without limitation.

Unfortunately, the reasoner, even with help of SPARQL queries, is not powerful enough to resolve restrictions in three-dimensional space. It means, for example, that the reasoner is not able to find out if there is a way how to place subparts together by the gripper of given sizes without collision. If we want to reach a solution, in which there is no need for the complete description of all skills and its sequence, that must be done to manufacture the product, there is a need for one more layer that can try to find a sequence of commands that would lead to the requested goal. All the articles mentioned before are gathering information for product assembly only from the ontologies. However, use of planner on top of ontology is not a novelty. In the field of autonomous robotics, there are many approaches [MPB+11], [BBM10],[MAR15] that connect ontology with a planner.

To test this approach for industrial use and to see the limitation of use of planner, the real production line will be used to see if it is the promising solution to the problem.The Testbed in CIIRC will be used to verify this approach. As the main robot the collaborative robot Kuka LBR iiwa will be used that can manipulate with position repeatability of 0,15 mm. To prove, that chosen solution will be the general and the right way to go, the planner will be tested on different more complex tasks first and then connected to the ontology and used to place electronic parts on a circuit board. To have sufficient environment suitable for this task, Stratasys 3D printer will be used to print parts as fingers for gripper or plates to fix the position of the products during the process of assembly.

1.2 Thesis structure

The thesis is organized as follow. A short introduction to the devices and tools used for flexible manufacturing are described in chapter 2. Purposed architecture, robot calibration, the design of supportive tools and implementation of knowledge representation are summarized in chapter 3. In chapter 4 are described tests for approach evaluation. In chapter 5 are described experimental results and potential improvements.

Chapter 2

Means for flexible manufacturing

The work itself was divided into two stages. The first stage was designed to proof the concept of use of planner in the chain of decision making, the second to connect whole chain. The equipment and architecture used during all phases are described below. The work done on those devices is described in next chapters.

2.1 Montrac

The Montrac monorail system is an intelligent modular automation and transport system that enables high scalability and flexibility thanks to separate shuttles, which rides on the rail, and thanks to no single point of failure[Gmbc]. The shuttles provide vibration free transport with safety sensor-controlled autostop function to avoid collision with possible obstacles[Groc]. Each shuttle can be controlled independently, so the products transported by one conveyor can be different and have different manufacturing processes. In the Testbed, which is situated in the new building of Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) is available one of those Montrac monorail conveyor systems. Testbeds conveyor system is equipped with three shuttles, that can park in three stations and as a whole is controlled by Siemens S7-1200 PLC.

2.2 Robots

The Montrac conveyor links three Kuka robots. One of the robots is collaborative robot LBR iiwa 14 R820, and two of them are compact Agilus KR 6 R1100 sixx.

Axis	Range of motion	Speed with rated payload
1.	+/- 170°	300°/s
2.	+45° to -190°	225°/s
3.	+156° to -120°	225°/s
4.	+/-185°	381°/s
5.	+/-120°	311°/s
6.	+/-350°	492°/s

Table 2.1: Axis data of KR 10 R1100 sixx

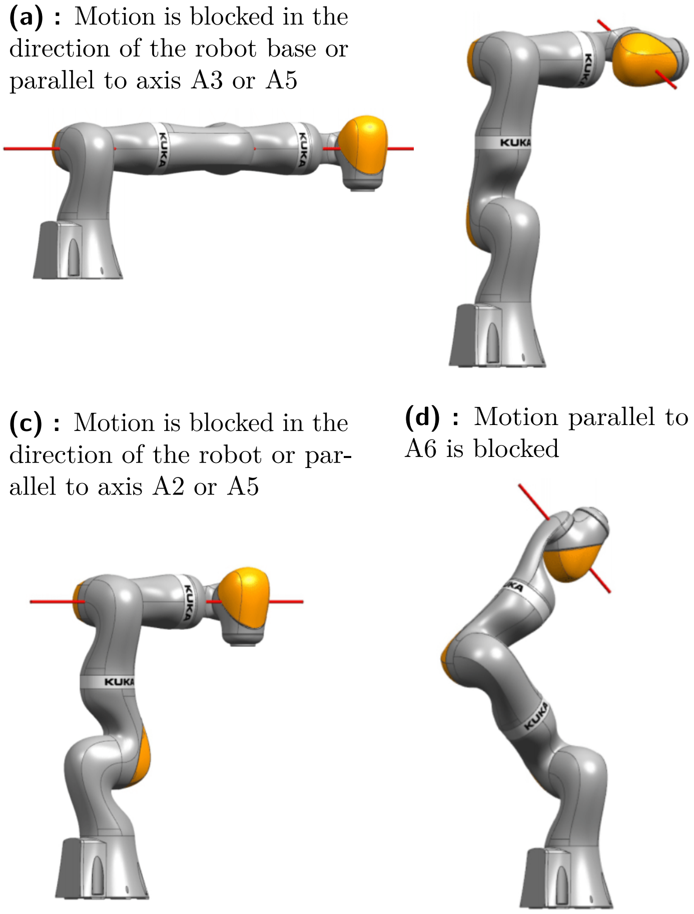
2.2.1 Agilus robot

The Agilus robot is a six-axis robot that achieves high precision movements (position repeatability is up to 0.03mm) irrespective of the installation position[Gmb15]. The robot was designed to be able to manipulate payloads up to 6 kg even in confined spaces. The Agilus robot is mainly designed for high working speeds and therefore is well suited for pick and place operations. Thanks to its lifetime lubrication, it never needs a change of lubricant in the gear units and therefore it has minimal maintenance requirements so it provides really continuous run. Kuka offers four variants of Agilus: standard variant, cleanroom variant, waterproof variant and EX variant, which is the waterproof variant with explosion protection[Gmba]. Robots available in the Testbed are standard variants of the Agilus robot. Range of the motion and speed in each axis is in table 2.1. Both Agilus robots are controlled by I/O system Siemens ET 200SP.

2.2.2 LBR iiwa robot

The LBR iiwa is a lightweight, human-robot collaboration capable robot. The iiwa robot was designed to meet all criteria of ISOs for human-robot collaboration, and therefore it can work in factory side by side with the human without any threats. All collaboration is possible due to joint torque sensors in each of the seven joints of robot. Thanks to it, it can reduce speed and force in joints whenever the robot detects contact, and behave as the joints are elastic. The torque sensor accuracy is 2% of axis-specific maximum torque, and position repeatability is 0.15mm [Gmbb], so the robot is ideal for precise cooperative manipulation.

The most significant advantage and the reason why iiwa was used as the main robot for testing scenarios are available connectivity and fact that it can be programmed in standard java without any limitations [Gmb16]. Java library for iiwa that contains classes and functions for robot control allows the programmer to use principles of object oriented programming, with the help of which even the complex algorithms can be easily programmed and maintain. Another advantage of iiwa is its high dexterity due to its seven joints. Thanks to 7DOF it can avoid an obstacle in a trajectory of the arm. But the high dexterity brings a disadvantage too. With more joints robot has more singularities and therefore more pleces to aviod. All of them are shown in the picture 2.1. The Kuka LBR iiwa is controlled through Profinet by I/O system Siemens ET 200SP, that is the same type as for Agiluses. Our LBR iiwa robot is equipped with the pneumatic flange that has two air walk-thoughts, EtherCAT connector, and three power jacks.



(a) : Motion is blocked in the direction of the robot base or parallel to axis A3 or A5

(b) : Motion parallel to axis A6 or A2 is blocked

(c) : Motion is blocked in the direction of the robot or parallel to axis A2 or A5

(d) : Motion parallel to axis A6 is blocked

Figure 2.1: There are 4 different robot positions in which flange motion in one Cartesian direction is no longer possible. Here only the position of 1 or 2 axes is important in each case. The other axes can take any position. (Source: [Gmb16])

2.2.3 Actuation

All robots are air actuated by the pressure of 6 bars that is available in the whole testbed. In case of iiwa, air pressure is controlled by the proportional vent. From the vent air is divided into selenoids that closees airflow for the robot and for locking mechanisms of Montrac stations. In case of Agilus robots, airflow is controlled by airflow control island, shown in the picture 2.2, thatis connected to the proportional vent too. All air control devices in the Testbed are made by Festo.

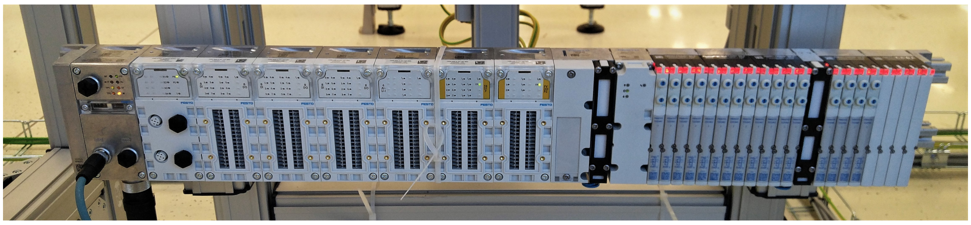


Figure 2.2: FESTO airflow control island

2.2.4 Grippers

Available grippers are air actuated and are from Festo too. All of them are shown in the table 2.2 on page 18.

2.3 Safety

As every production line, even ours has to follow security requirements that are described by ISOs. In case of Agilus robots, that are not equipped with any sensors that detect colision, the intrusion of human must be observed by a detection system. In our case, safety laser scanners S3000 PROFINET IO Advanced from Sick is used as the detection system. S3000 is a scanner that can provide two user-defined fields at once. The scanner can provide warning field under the distance of 49m and the protective field with resolution up to 30mm under the distance of 4m. The response time of scanner is 60ms.

Described information, as the speed and the position are all the information needed to compute exact requirements for given application for ISO 13855:2010. In case of collaborative robot that iiwa is, the limit requirements for the impact forces on different parts of the human body are given by ISO 15066:2016. In applications for industry 4.0 can be hard to compute and change safety devices whenever manufacturing process change, because of the different, customized products. Therefore, it is easier to take into account worst-case scenario from all applications and compute requirements from that.

2.4 Communication

All devices on the Testbeds production line communicate over Profinet. Profinet is industrial ethernet based on experiences with Profibus. It is fully compatible with regular ethernet communication but has some advantages, for example real-time communication that works thanks to masters that control data flow. Because Profinet can control each data input/output differently, there is need to set profiles that configure Profinet control functions following requirements of environment.

2.5 3D Printer

For different applications, different tools as grippers or holders are needed, and that is the time when an industrial 3D printer can come in handy. In our case, the additive printing technology is provided by 3D printer Fortus 450MC from Stratasys that enables prints with an accuracy of 0.1mm with the technology of Fused Deposition Modeling (FDM). The printer is also able to print parts from two materials simultaneously which can be very useful. When one of chosen material is special support material that can be wash out from the product, printer enables to print movable parts as gearboxes for example. Another feature of this Stratasys printer is the possibility to print in non-thermoplastic materials as are nuts, bolds, or whole circuits.

2.5.1 3D Printing

3D printing is Additive Manufacturing (AM) technology that allows to the material to join or solidify in a controlled manner to form three dimensional objects. This technology is already being adopted for rapid prototyping and soon for rapid manufacturing.

There are five stages of AM that most models has to pass through until it is finished. Before printing, the product needs to be modeled in computer-aided design (CAD) program such as Autodesk's Inventor, Dassault Systèmes's SOLIDWORKS, or Siemens's NX. Those programs aloves to test the mechanical properties of the model before it is printed and therefore reduce costs. When a CAD model of the product is done, the product needs to be exported into the STL file which is the standard file format for 3D printing. In this format, the model is described as a bunch of unstructured triangulated surfaces without any information about color or texture. As a next step, the STL file needs to be imported into a software, mostly made by the 3D printer manufacturer, that prepares the model for printing itself. In most cases, the program is a slicer, which slices model into separate layers of print. After that, the model is sent to the printer where it is printed. As the last stage, 3D printed model needs to be cleaned and polished. All stages are shown in the picture 2.3).

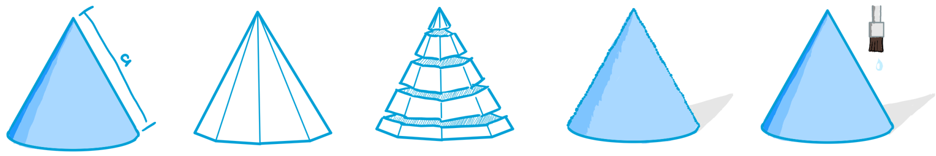


Figure 2.3: five stages of 3D printing

As the technology of 3D printing is here for more than 30 years, many processes of printing are available. Those processes can be sorted into seven groups.

- **Vat Photopolymerization** is a process during which a photopolymer resin is exposed to the light of a specific wavelength that causes solidification. The mechanical anisotropy for Stereolithography which is a member of the group is around 1%.
- **Powder Bed Fusion** is a process which produces solid parts by sintering or melting powder by the thermal source. For Selective Laser Sintering as a representant of the group is mechanical anisotropy around 10%.
- **Material Jetting** is the process where drops of material harden when exposed to UV light or elevated temperatures. This process is similar to the 2D ink jetting.
- **Direct Energy Deposition** is the process used almost exclusively in metal 3d printing. This process uses an electron beam or laser that resolidifies alloy on which powder or wire is implemented.
- The next group of printing technologies is **Laminated object manufacturing** where a sheet of heat-activated material is rolled out and laminated onto previous layers. The sheet is then cut by laser.
- "**Binder Jetting** deposits a binding adhesive agent onto thin layers of powder material. The powder materials are either ceramic-based (for example glass or gypsum) or metal (for example stainless steel)."[HUB]
- **Fused deposition modeling (FDM)** is process patented by the co-founder of Stratasys in 1989. It belongs to Material extrusion group of

> AM technologies. "Extrusion-based AM generally follows the printing principle of extruding material and depositing onto a platform creating a two-dimensional layer on top of another resulting to a tangible three-dimensional object. Among other extrusion-based techniques, FDM is a material-melting technique which uses a spool of thermoplastic filament such as PC, ABS, and PLA with varying diameters to be melted and extruded through a heated nozzle."[DECA18]

This lastly mentioned technology is the most widely used technology of 3D printing. It can produce prototypes in short amount of time and can be cost-effective. The disadvantages of technology is dimensional accuracy limitations and anisotropicity. Fortunately, in some cases, dimensional inaccuracy can be avoided. Because of inaccuracy of printing, smooth threads are impossible to be printed but can be avoided by imprinting the nut into the model. The big problem is the mechanical anisotropy that cannot be avoided. The mechanical anisotropy of FDM is the largest among all additive manufacturing techniques. The Article [CCGPN17] shows that during tensile stress test performed following the standard ASTM endurance of part printed in the upright position is up to 4 times lower than the same part printed in the flat position. The results of flexural stress tests were similar. The endurance of part in the upright position was up to 4 times lower than the same part in the on-edge position. The only difference was that to endure flexural stress it is better to print parts in the on-edge position whereas to endure tensile stress it is better to print parts in flat position. The stress-strain graphs from tests and print positions of the tested part are shown in the graph 2.4 on page 14 and graph 2.5 on page 15.

Except for anisotropy, there is one widespread defect connected to FDM. When the material is cooling down, its dimensions are decreasing, and model deform itself. To reduce the probability of warping many printer manufacturers are offering heated build-platforms or even heated chambers. From the designer point of view, the probability can be lowered by avoiding large flat areas, avoiding sharp corners, or by not printing from different materials especially those with different coefficients of thermal expansion.

2.6 Knowledge Representation

2.6.1 Ontology

The Ontologies are structures for categorizing and classifying data, its relations, and properties with given syntax and logic. Those structures are mostly designed to describe the structure of knowledge for a particular domain of interest, so the structure can be more precise than if it would be designed for

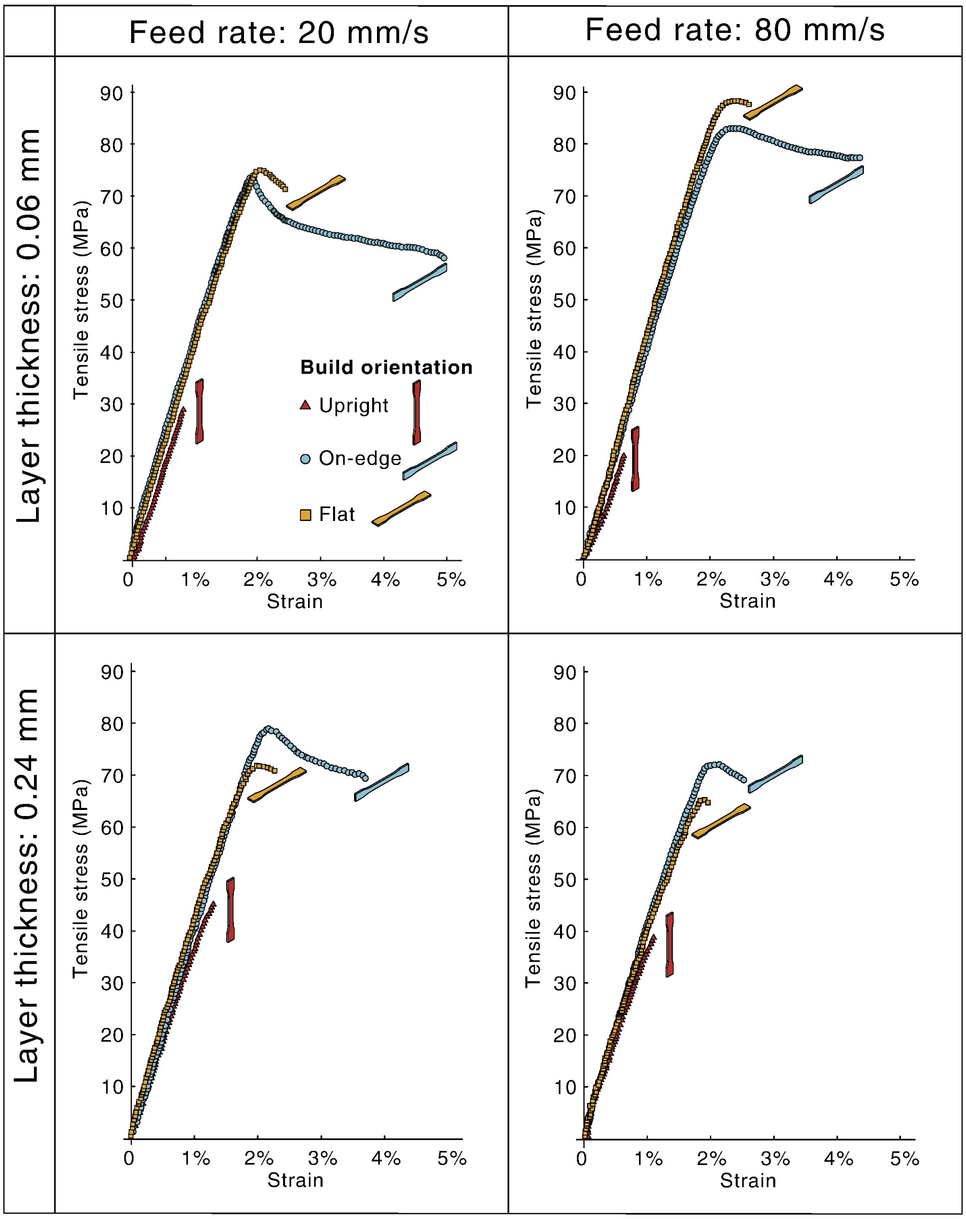


Figure 2.4: Tensile stress test (Source [CCGPN17])

the general domain. One of the most used languages for ontology description is family of Web Ontology Language (OWL) exactly OWL family and OWL2 family, where OWL2 is extended version of its predecessor[Groa]. The OWL languages are languages with formal semantics that is built upon XML standard for Resource Description Framework (RDF) made by World Wide Web Consortium. That is the reason why RDF/XML syntax is the only mandatory syntax for all OWL tools. Other possible syntaxes are OWL/XML which makes it easier to serialize using standard XML serializers, Functional Syntax which makes it easier to see the formal structure of ontologies, Manchester

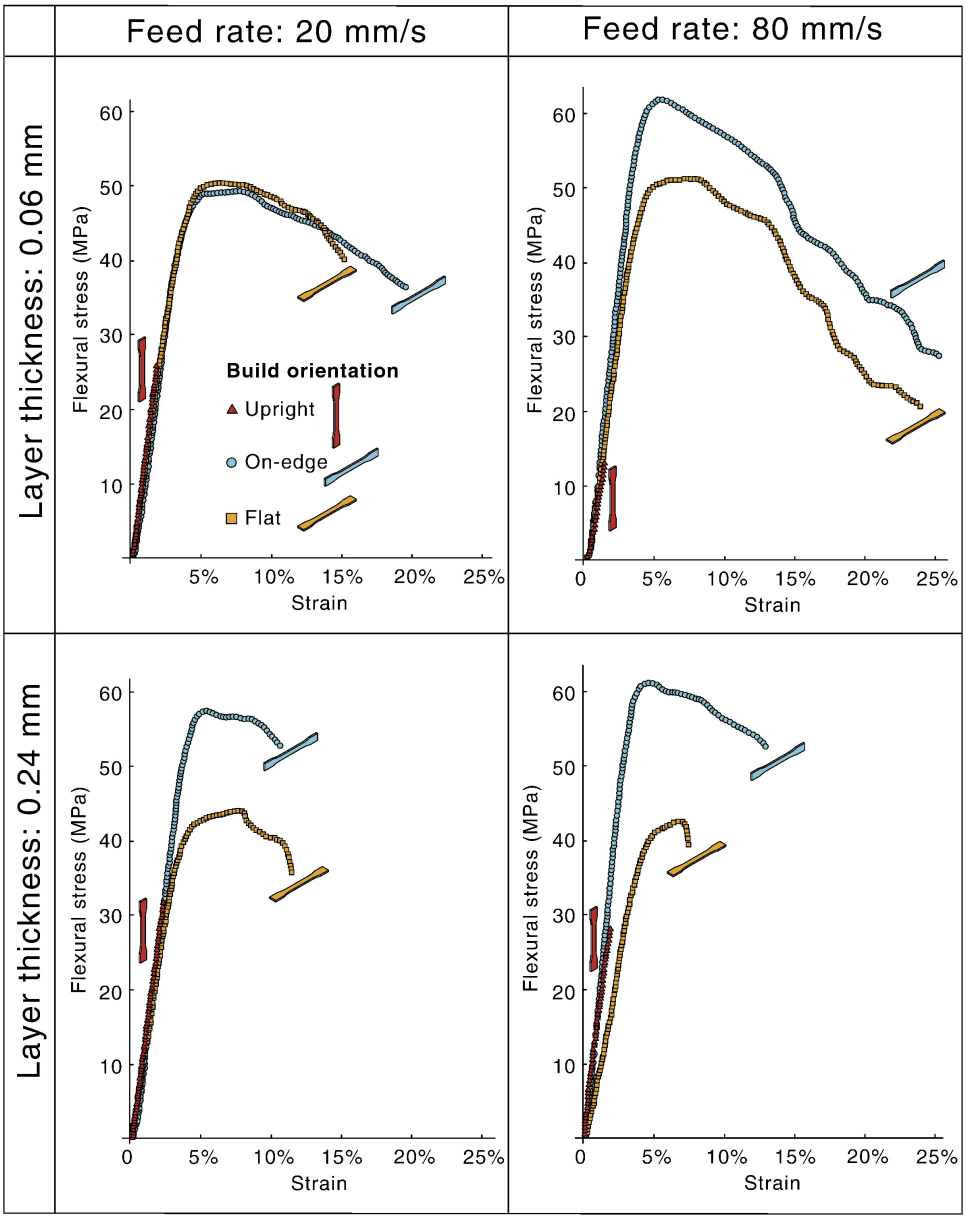


Figure 2.5: Flexural stress test (Source [CCGPN17])

syntax which makes it easier to read and write DL Ontologies and Turtle syntax which makes it easier to read and write RDF triples.

Independently on syntax, there are two types of semantics that provide two alternative ways of assigning meaning to OWL ontologies. The first type of semantics is Direct Semantics sometimes referred to the ontology that uses it as OWL DL ontology[Grob]. OWL DL is designed to be compatible with the model of the theoretic semantic of the SROIQ description logic that is the fragment of first-order logic which properties can be utilized in most reasoners.

The second type of semantic is RDF-based Semantic sometimes referred to as OWL FULL and is fully compatible with the RDF semantics. For most cases, more advantageous semantic is OWL DL because of useful properties for reasoners. All of described semantics and syntaxes are shared in both OWL and OWL2. The functionality that OWL 2 adds are for example richer datatypes and data ranges, a disjoint union of classes, and many more.

The reasoner itself is reasoning engine that derives logical consequences from a set of facts or axioms. Most of those reasoning engines are based on tableau calculus that uses truth tree as a proof procedure. The tree consists of given set of formulae that are then, with use of Tableau rules, expanded into branches[Nie96]. If all branches of the tree are closed, that means that contradiction occurs in that particular branch, given set of formulas is unsatisfiable. If there is a branch that is closed and there is no Tableau rule that was not used, set of formulas is satisfiable. In case of this thesis, description of the product can be in separate ontology where all tools necessary for manufacturing are listed, and description of the available tools can be in another ontology. Reasoner than, with use of SPARQL protocol, enables to check if the factory can make the described product.

SPARQL is an RDF query language, that is able to retrieve and manipulate data stored in RDF structured. SPARQL allows to user to use the full set of query operations to search the entire database[Groe].

This solution is sufficient, but there is still need to have all tools that are necessary for manufacturing process and sequention of opperations listed in product description. To avoid this complex set of information and to make the description of the product more straightforward and more general we can try to plan manufacturing process only from a description of the product and its parts, and from a description of capabilities of the production line. For this purpose, the reasoner is not sufficient solution because it is not able to take into account constraints of the 3D world.

2.6.2 Planner

The solution that can eliminate the problem is to link information into planner which can plan the process. The planner is an engine that based on initial states, and possible actions generate, if possible, a sequence of operations that leads to the predefined goal. If the environment is known and models are available planning can be done offline. On the other hand, if the environment is dynamically changing or unknown, planning must be done online based on a sensory or other input data. Both offline and online planners enable generating a sequence of actions even for complex tasks in multidimensional space which is the property that makes it ideal for electronic parts assembly and for joining the subparts generally. For the description of planning domain, there are many different languages.

In 1998 there was an attempt to standardize planning domain languages and based on STRIPS and ADL, the new language called Planning Domain Definition Language (PDDL) was created. The same year was organized first International Planning Competition (IPC)[Hela] in conjunction with the fourth international Artificial Intelligence Planning and Scheduling conference. The outcome of that competition was the adoption of PDDL as a common representation language for planning. Since then as an output of IPC events, new features of PDDL were introduced. The last version of PDDL language is PDDL 3.1 and supports numeric and object fluents, derived predicates or soft constraints. Numeric and object fluents are functional state variables to model non-binary resources which means that numbers, mathematical operations and comparison of numbers can be used. Derived predicates are state variables which are computed as a function of other state variables, and soft constraints are constraints that need not be satisfied by a plan but lead to a decrease in plan quality if they are not. [FL03], [GL05]

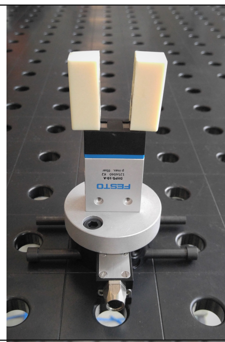
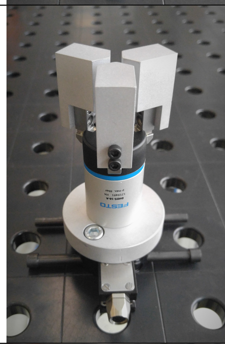
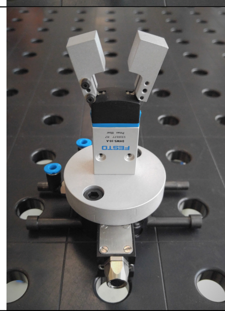
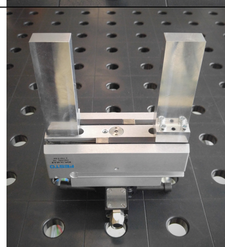
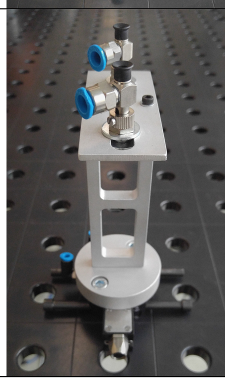
Name	Stroke* / Diameter	Accuracy**	Picture
DHPS-10-A	3 mm	0.02 mm	
DHDS-16-A	2.5 mm	0.04 mm	
DHWS-10-A	20°	0.04 mm	
HGPL-14-40-A-B	40 mm	0.03 mm	
Suckers ESG	8 mm (diameter)	—	

Table 2.2: Available FESTO grippers and suckers (* Stroke is per jaw of gripper)(** Position repeatability accuracy)

Chapter 3

Design & Implementation

As described in the introduction, many researchers exploit ontologies for an easy robot programming. These researchers are using structures of skills (some kind of elementary abilities) to describe tasks for the robotic unit. The robot is programmed only with the use of skills, so the programmer is constructing program only with the use of these functional blocks. When for example the robot needs to be reprogramed for new pick and place operation programmer is editing only "pick" and "place" skills instead of dealing with for chosen robot native programing language. If the existing skill is not sufficient programmer can change functional block and create new skill only by changing subskills. The product then can be described by skills that need to be done and reasoner can check if all of the specified requirements are met. The approach like this makes things easier but leads to another problem. The reasoner is not able to decide if one part can be assembled before another just from the properties of those parts. Therefore the product needs to be described with the exact sequence of operations in mind.

However, this problem has a solution. It can be solved by adding one more decision level. After the reasoner checks that all skills are available the description of the product can be sent to another layer where a planner can search state space if there is the sequence of operations that leads to the goal. Use of planner brings more advantages. The planner is capable of going through all possible sequences to find the optimal one, in case that there is a finite number of possibilities. If not it can find the best solution from a limited number of possibilities or one that overperforms some given threshold. Whenever the manufacturing process based of this solution fails, because of for example faulty device, the process can be replanned to avoid that device if possible.

In this chapter will be described an attempt to implement such a solution.

3.1 The architecture of decision making

Proposed architecture of the decision making is shown in figure 3.1. At uppermost layer, there are three ontologies. The main ontology describing capable skills of the robotic unit, another describing available or possibly available parts of the store and input one, that describes desired skills for the product. This architecture can be used during initial phase to check ability to manufacture a given product, and the same one can be used during the manufacturing process itself to monitor if all needed subparts are available or if error occured. The main and store ontology are separate based of idea of different frequencies of an update. The store ontology should be updated everytime the subpart is taken, but the main ontology should be updated only when there is a change or error in the available toolset. Another reason is error resistance and robustness. When there is an error in the storing unit program, it should not be able to influence the main ontology. Thanks to this architecture the storing unit can be restricted to operate only with store ontology.

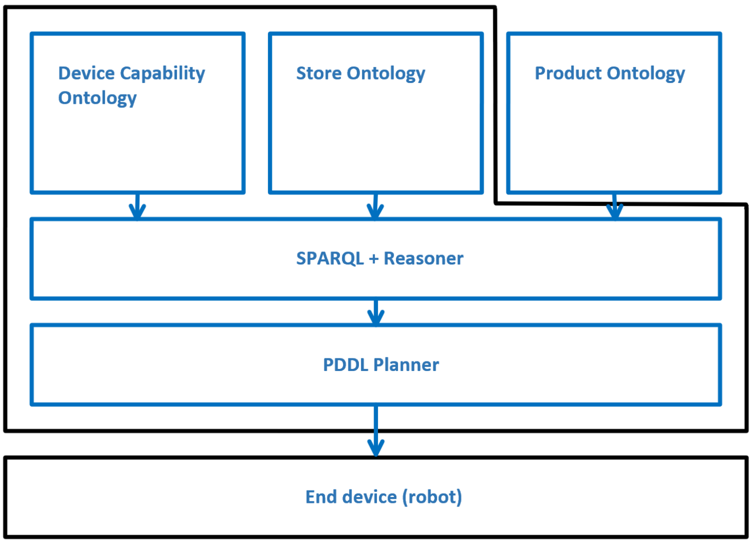


Figure 3.1: Diagram of the architecture of decision making

In next layer, there is semantic reasoner and SPARQL queries for the prooving availability of skills. When the skillset is proved, the next step is planning.

Because in manufacturing industry there is the strong emphasis on the robustness of the solution the PDDL language is chosen as the necessary requirement for the used planner, as it is used and tested for about twenty years and it is defacto standard in planning languages. If this PDDL planner finds a satisfactory sequence of operation, the result is sent to the robot in the form XML file. The robot then executes the sequence of operation in given order.

In contrast with [Ste15], in this approach the code for every skill is already stored in the robot. The unit then executes the functions corresponding to the sequence of elementary skills. The idea behind not to have universal code for each skill stored on the server is that setup of each robotic cell varies a lot. On today's market, there is a tremendous amount of sensors, grippers, PLCs and other device and tools that robot have to work with. Therefore it is almost impossible to provide code for every possible combination. As a better solution seems to be definition or standardization of every elementary skill and every unit that should provide particular skill should meet the standard. With this approach, every unit can be optimized for specific realization and utilize all its possibilities. Ofcourse there is a distinct disadvantage of this approach. During the first implementation of such a unit, all the programming needs to be made to measure, so expert is needed (perhaps with use of snipets and frameworks). But after that unit should be flexible and change of manufacturing process should be low-cost.

3.2 First stage - Capabilities and limitation of PDDL planner

To prove suggested architecture and to see capabilities and limits of each part it seems like the best workflow to make a minimal solution and then add technologies iteratively. Because there is the desire to find out capabilities and limits of the present PDDL planners turned into the manufacturing process the goal of the first stage is to prove cooperation planner-robot on some spatially more complex task. After a discussion, as such a task, was chosen LEGO assembly. This is because all three dimensions of space must be taken into account and because it is similar pick and place operation as circuit boards assembly so there will be a need for only small adjustments.Another reason was that there are many projects that can utilize the outputs of successful implementation and that there is already working solution for connecting LEGO Designer to PDDL planner made by RNDr. Jiří Vyskočil, Ph.D.

The concept of proofing was as follows. First to program and set iiwa, then connect iiwa to the PDDL planner and define problems of such an implementation or setup. After that, solve if it is possible, problems from the first stage and connect planner to ontologies instead of LEGO Designer. Following this scenario, more problems can be found, and the solution can be therefore more general and better proofed.

3.2.1 Repetability tests of LBR iiwa

From previous work with LBR iiwas, there were concerns about the accuracy of the robot. Other iiwas that were used on other projects were showing signs of wear as they were used on many applications during the time. Those signs of wear reduced repeatability accuracy, so the movements along some axes were not straight. This error in displacement between two points, that were distant one meter from each other, was more than one centimeter and therefore this robot had to be maintained by the KUKA.

To avoid affecting of results by uncertainty, two tests were made on LBR iiwa. The first test was made to prove ability to follow the straight line. The available welding tables that robot is mounted to have a grid with a five centimeters wide frames. Because this grid system is designed for precision clamping, it was used as the reference. In the grid were chosen three points. One point was placed at the intersection of two perpendicular lines of the grid and two points in one-meter distance from the origin, each on a different line. Those points are shown in the picture 3.2. The task for the robot was to move repeatedly along those lines defined by three points in an L-shape movement. The needle was attached to the end-effector to have reference of position. For movement along straight lines was chosen linear movement function of LBR iiwa.

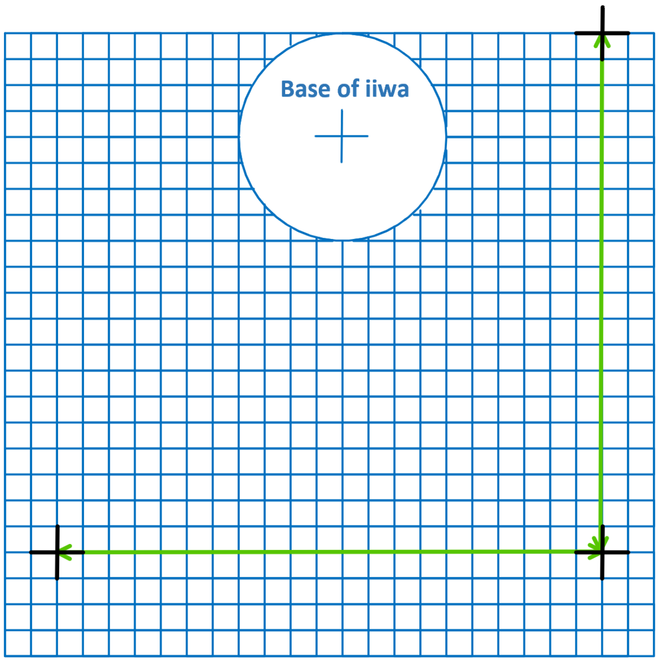


Figure 3.2: Diagram of the test. The black crosses are the three chosen points, the green arrows are the trajectories of movement. Size of each frame of the grid is 5 cm

During the whole test, the iiwa robot was moving the needle in the one-millimeter thick grove of the grid, so the test was closed as successful because there werw no signs of unaccuracy.

The second test was made to invalidate the idea of inaccuracy of repeatability during the long-range time that could be caused by warming up. As there was a need for some kind of record of progress during the time and there was no camera with stative to record the progress, approach for this test was chosen as follows (setup is in the picture 3.3). A 0.1 mm thin liner pen was attached to the gripper. With use of previously defined three points, thebase that is parallel to the desk of the table was defined. The program for the robot was written to make dots on two papers of the A3 format in distance of 3 cm in the x-axis and 4 cm in the y-axis. In between, each two points robot have to move in all joints to warm up all motors. The test had been run for over 18 hours.

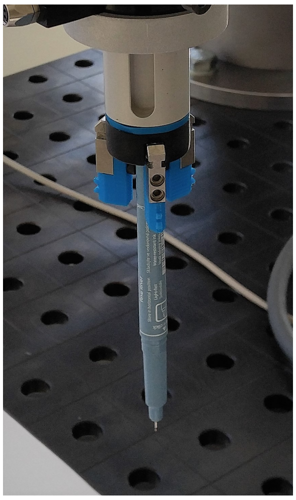


Figure 3.3: The setup for the second test.The FESTO DHDS-16-A gripper with the linker pen

After the test, the dots were evenly distributed with no distinctive difference in size of spots and with no signs of inaccuracy along the surface of whole paper. After the test, the dots were bigger than pen tip, approximately 1 mm in diameter. It was assumed, that it was caused by the significant amount of ink released from the pen during the time. This assumption was based on the last cycles of test, during which the robot was dotting precisely to the center of the spots. The test of the accuracy of the warmed up robot was therefore closed as successful. Part of result of the test is show in the picture 3.4.

3.2.2 Calibration

After it was proofed that there is no significant error in accuracy, there was the need to calibrate tool frame. Because the tool was mounted on a custom-made middle-plate that had an unknown angel of the mount the calibration

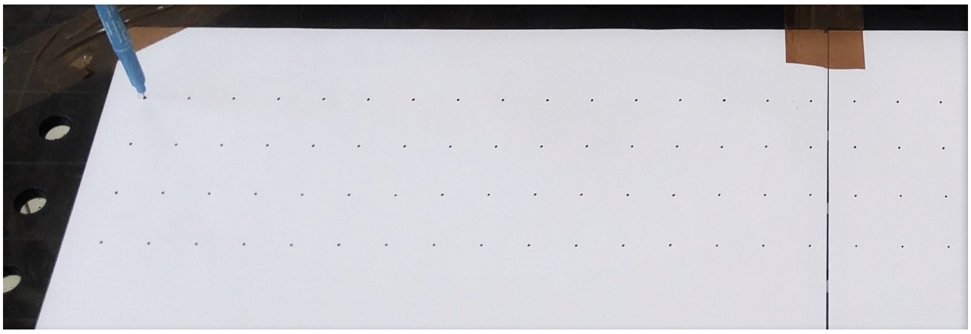


Figure 3.4: Part of the result of the second test

was necessary. Calibration had to be done so the direction of x-axis was going through the center of both fingers and the z-axis was perpendicular to the surface of the flange. The wanted direction of the tool frame is shown in the picture 3.5.

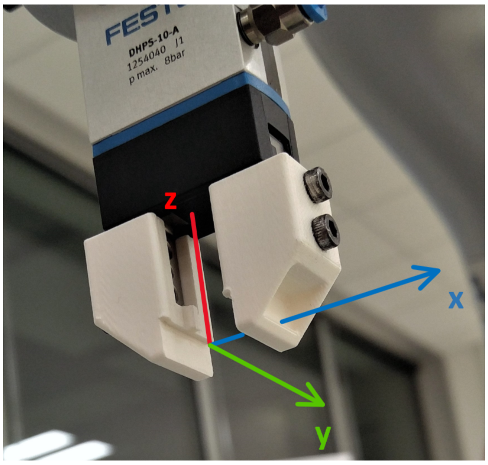


Figure 3.5: The wanted orientation of the tool frame

Kuka LBR iiwa offers four jogging types and for each type seven options of movement to move manipulator. Modes and options are shown in the table 3.1. To make things easier iiwa offers methods for tool and base calibration. Those methods can define tool orientation and transformation but with for application insufficient accuracy, because they strongly depend on the estimate of a programmer. Therefore, the output of the calibration needs to be refined manually and for such a tool calibration, it is most advantageous to use Tool jogging type.

The exact procedure is as follows. First is done the raw calibration with use of calibration tool called ABC World. When the gripper is moved to the position where the orientation of world base approximately corresponds to the orientation of wanted tool frame ABC World can be used to assign world base orientation to the tool frame. Then the gripper needs to be rotated so

Type	Description	Options of movement
Axes	The robot is moved by axis-specific jogging	Around 1st-7th axes
World	The selected TCP is moved in the world coordinate system by means of Cartesian jogging	Along or around x,y,z axes + redundancy
Tool	The selected TCP is moved in its own tool coordinate system by means of Cartesian jogging	Along or around x,y,z axes + redundancy
Base	The selected TCP is moved in the selected base coordinate system by means of Cartesian jogging.	Along or around x,y,z axes + redundancy

Table 3.1: Table of jogging options. By redundancy is intended the movement of manipulator body with the tool set at the same position. This is possible because of the redundant joint. (Source [Gmb16])

the edge of the tool is parallel to the groove of the welding table grid. If the gripper is set to the Tool jogging mode and is following the lines of the grid when moving along axis, the calibration is done. If not, values of rotation of gripper have to be changed manually in Sunrise Workbench (programming environment for LBR iiwa based on Eclipse) until it is calibrated well.

The procedure for orientation calibration of the base-plate was similar. At the beginning of the process, the orientation of plates base-frame was roughly calibrated by Three point method. Then, by using Base jogging method, the base frame was modified in Sunrise Workbench until the movement along x-axis and y-axis were parallel to the edges of the plate. After orientation, the exact position had to be set, and for the exact position, the correct approach needs to be found.

To avoid singularities and movements near to them, shown in the picture 2.1, the position of the workspace needs to be well placed, and the manipulation space needs to be constrained. The problem with movement near singularities is, that during the motion, joints needs to change its orientation rapidly, almost in instant time, and because of limitation of motors the precomputed trajectory change its dynamics and slows down.

From the considered variants, the best approach seemed to be to set the workpiece frame into the center of two by two pins brick. The chosen gripper was FESTO DHPS-10-A thatr was designed, so it is capable to grip only brick two pins wide. Exact reasons for this setup will be described in the section of the design and 3D print. The reason for placing workpiece frame in the middle of 2x2 brick is, that with a ninety degrees rotation around the center, only orientation is changing and position stays the same. This means that it is possible to place arbitrary rotated brick with the same transformation coordinates and without difficult computations. If the gripper would be capable of gripping brick of different widths, the better approach would be to set the workpiece frame into the center of some pin, so with every size of a brick and every rotation the brick could be placed.

The problem that has occurred during calibration was that even with symmetric gripper holding the brick precisely in the middle of the brick, the center of the brick was slightly shifted. Reasons for that were slightly larger holes for the mounting bolts of the gripper that allowed shift during tightening. To compensate this shift, the tool had to be manually recalibrated. The method used for calibration was to twist the brick 180 deg around its center. After the twist, the difference in x and y displacement of the center of brick before and after rotation was exactly two times displacement of brick center (shown in the picture 3.6). By repeating this process, the displacement was minimalized to the negligible value. To proof the correct position, following sequence of operations was used:

```
move to position of brick;
close gripper;
take brick;
turn 90 degrees clockwise;
place brick;
take brick;
turn 180 degrees clockwise;
place brick;
take brick;
turn 270 degrees clockwise;
place brick;
take brick;
turn 0 degrees clockwise;
place brick;
open gripper;
```

When the brick sat precisely on pins after all turns, the calibration was successful. This approach was used to set base of three available plates. After calibration, the brick could be arbitrarily transported between all base-plates.

3.2.3 connectivity

The LBR iiwa Sunrise cabinet is equipped with two ethernet connectors, the X11 connector for safety devices, the DVI monitor output and with several USBs. The iiwa used in the Testbed was already connected to PLCs by Profinet connection, so only one ethernet connector was free. In default setting, the second Ethernet connector is set for KONI/FRI interface.

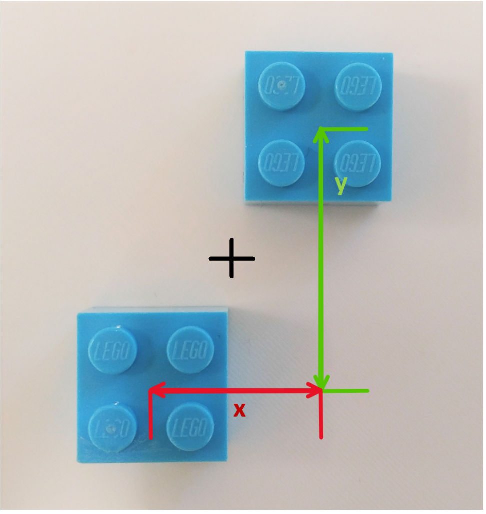


Figure 3.6: Method for measuring of displacement. The black cross is displaced center of brick. The red value is displacement of center of brick before and after rotation along x-axis. The green value is displacement of center of brick before and after rotation along y-axis.

"The Fast Research Interface Library runs on a remote PC node with is connected to the KRC (KUKA Robot Controller) via an Ethernet connection. In intervals of 1 to 100 milliseconds, UDP packages are periodically sent from the KRC unit to the remote host. These packages contain a complete set of robot control and status data (e.g., joint positions, joint torques, drive FRIDriveTemperatures, etc.; cf. FRI User Documentation). The remote host (e.g., with QNX Neutrino RTOS) has to instantaneously send a reply message after the reception of each package. A reply message contains input data for the applied controllers (e.g., joint position set-points, joint stiffness set-points, etc.). This way, users become able to set-up own control architectures and/or application-specific controllers for the light-weight arm as it is often desired at research institutions."[Grod]

The Profinet profile was already set and used for other projects, and without changing its properties, it was impossible to send data packages through. Because of that, the KONI interface had to be changed, so it communicates with java application running on Windows through TCP/IP communication. By following this process, the KONI interface was mapped (Because there was only one webpage[Vir] with these instructions, I am listing short version of it for case of deletion):

1. Connect mouse, keyboard and monitor and log-in into the cabinet. (The required password can be found via Google)
2. Stop KRC program in the system tray. (green icon)
3. Open the list of the network interfaces: Start -> View network connections. At this point, you should only have one interface: Realtime OS

Virtual Network Adapter.

4. Run in the terminal to assign KONI to Windows:

```
C:\KUKA\Hardware\Manager\KUKAHardwareManager.exe -assign
    OptionNIC -os WIN
```

5. Reboot cabinet and change the IP address of newly created ethernet adapter, so it is in a different subnet than the other one.

6. Reboot again

- To revert changes run in the terminal :

```
C:\KUKA\Hardware\Manager\KUKAHardwareManager.exe -assign
    OptionNIC -os RTOS
```

More detailed instructions can be found here [Vir]. To verify the communication, a simple java TCP/IP echo server for iiwa and client for pc was created. There were concerns that Ikarus Antivirus running on the iiwa cabinet can block the communication but those concerns have not been confirmed and communication was succesful.

3.2.4 Design of fingers for grippers

The requirements for the gripper were following:

- The gripper has to be able to take and place brick 2x2 and 2x4 pins on LEGO board.
- The gripper has to be able to take subpart constructed from three bricks (two brick at the bottom, one on the top).
- The gripper has to resist the closing pressure 8bars.
- The gripper should be able to disassamble product piece by piece.
- The gripper should fit as small place as possible to have as few constraints as possible for products
- The gripper should provide as great precision as possible.
- The gripper should provide as great variance in parts that can be gripped as possible.

From available FESTO grippers, the three-point gripper was rejected immediately from the beginning. After analysis, the angle gripper and the large parallel gripper were denied because they occupied to much space in the open state (All grippers can be only fully opened or fully closed) and suckers had a too large diameter to fit in between pins of a brick so the DHPS-10-A have been chosen as the best candidate. With three millimeter stroke per jaw was possible to design fingers that would be able to operate in space that would correspond to two 2x2 bricks each placed from the one side of the transported brick. The disadvantage of this gripper was that it could be designed only for one width of the brick as the stroke of both jaws in total is smaller than 8 mm (size of one pin brick).

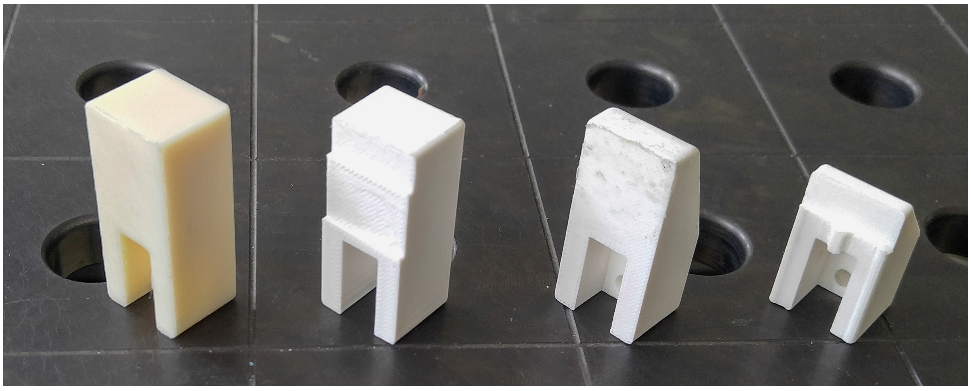


Figure 3.7: All versions of fingers for gripper. The first version is the first on tthe left. The latest third version is finger on the right.

The first goal for the first prototype, shown in the picture 3.7, was to test if the surface of finger covered by a rubber-like material is offering enough friction to allow placing a brick on a LEGO plate. The cover for finger was made by applying ten layers of Chemopren glue (glue for flexible joints) that creates a thin layer of silicon-like material. Because friction was too low, the second goal was to design way how to place and remove a brick. For brick placing, the protrusion on the finger was created to press the brick from the top. For the brick removal was designed movement, where a brick was released from pins by rotation around its bottom edge 3.8. Because of this movement, the side of the finger had to be chamfered to not colide with the base-plate. The holes for nuts were made realy tight to lower inaccuracy.

The second prototype shown on picture 3.7 was designed to proof those concepts and to show if it is advantageous to have two protrusions, one for subparts and one for single bricks. The second Prototype showed that it is possible to place and remove a part with use of described approach. The two protrusions were found unnecessary because it was possible to place subpart by holding only the top brick. Two protrusions had another disadvantage, and that was offset of the center of brick along the x-axis for the subpart.

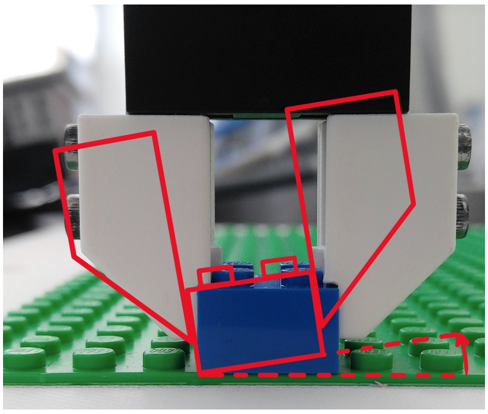


Figure 3.8: Method for removing a brick by tillting arount the bottom edge.

The third prototype shown on picture 3.7 was made based on that knowledge. Because of only one protrusion, the finger was designed shorter, to have higher stiffness. To eliminate the problem with offset finger was designed symmetrical and to increase accuracy tooth for positioning of brick was added. This tooth fits precisely between pins of brick, so the center of brick is every time in the same position. To lower amount of used material small depression was made at the top of the finger. Drawings of all prototypes are in the appendix (address).

For every prototype, the Finite Element Analysis (FEA) was made to ensure strength accuracy and permanence of fingers. The air pressure for which the FEA was computed was 6 bar per jaw of the gripper that is equivalent to 40N[FES17]. As the main criterium for structural integrity of fingers was chosen Factors of Safety (FoS).

FoS is defined in Merriam-Webster as "the ratio of the ultimate strength of a member or piece of material (as in an airplane) to the actual working stress or the maximum permissible stress when in use." [MW]

This ratio can be computed as follows:

$$FoS = \frac{\text{yield stress}}{\text{working stress}}$$

Where the yield point is the point on a stress-strain curve that indicates the limit of elastic behavior of the material. After that point, the material starts to deform plastically. Working stress is maximum stress on the product during normal use. The limit of FoS that was set as the minimal requirement was factor equals two. Another useful information that came from FEA was displacement caused by working stress especially when fingers were not symmetrical, von Mises stress and 1st principal stress.

"Von Mises stress is a value used to determine if a given material will yield

Version	Material	FoS	Von Mise	1st P.	Displac.
V1	ABS-M30	3.384	5.910 MPa	4.686 MPa	0.056 mm
V2 - f. finger	ABS-M30	3.154	6.341 MPa	6.868 MPa	0.060 mm
V2 - ch. finger	ABS-M30	2.752	7.267 MPa	7.97 MPa	0.062 mm
V3 - original	ABS-M30	3.31	4,897 MPa	6,041 MPa	0,022 mm
V3 - generative	ABS-M30	2.704	5,894 MPa	7,395 MPa	0,035 mm
V3 - manual	ABS-M30	3.075	7,391 MPa	6,504 MPa	0,041 mm

Table 3.2: Results of FEA. f. finger - flat finger, ch. finger - chamfered finger, Von Mises - Von Mises stress, 1st P. - 1st Principle stress, Displac. - Displacement

or fracture. It is mostly used for ductile materials, such as metals. The von Mises yield criterion states that if the von Mises stress of a material under load is equal or greater than the yield limit of the same material under simple tension, then the material will yield." [Sim]

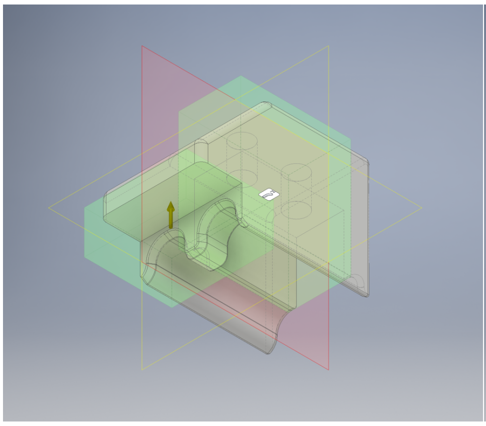
"The 1st principal stress gives you the value of stress that is normal to the plane in which the shear stress is zero. The 1st principal stress helps you understand the maximum tensile stress induced in the part due to the loading conditions."[Inc]

Maximum tensile stress for ABS-M30 made by Stratasys is 31 MPa. All values of stress analysis are shown in the table 3.2.

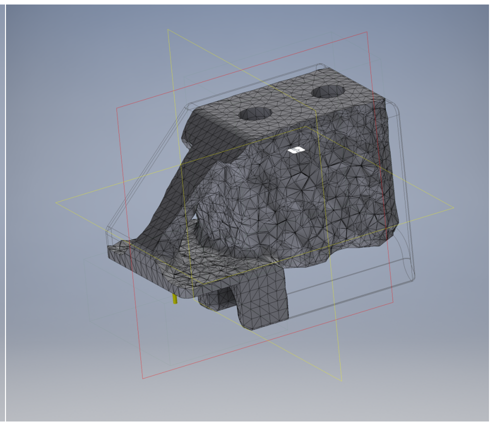
After the prototyping the third version of the finger was optimized for the amount of material used. First, the model was optimized manually and FAE was made to see differences from the original model. Then generative design tool was used to generate an optimized solution with use of AI 3.9b. This generated solution was then redrawn and analyzed 3.9c. (It ought to be mentioned that version of Inventor that was used was 2018 and newer version already exists. Especially in the new version of Fusion 360 is a whole new set of tools and functions for generative design). The generative design is new way of how to design things by seting conditions and constrains and leting the design on AI. For example Airbus was able to optimize cabin partition with this technology, so it is half of the weight of original one and even stronger [Con]. From the point of view of FoS, the manual solution outperform generative designed one and therefore was chosen for print, but the manualy made finger will suffer more from worn out.

As shown in a previous chapter, the direction of print can make a big difference in strength of the material. To maximize strength, the finger was printed in the flat position (following the chart 2.4), with gripping surface laying down to minimalize support use.

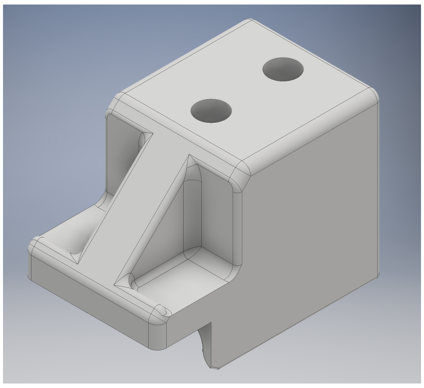
(a) : Generative design constrains. The red plain is plane of symetry. Two green boxes are preserve regions.



(b) : Result of the generative design generated by the Inventors artificial inteligence.



(c) : Redrawed model of generative optimalization.



(d) : FoS set as safety factor. One of results of FEA.

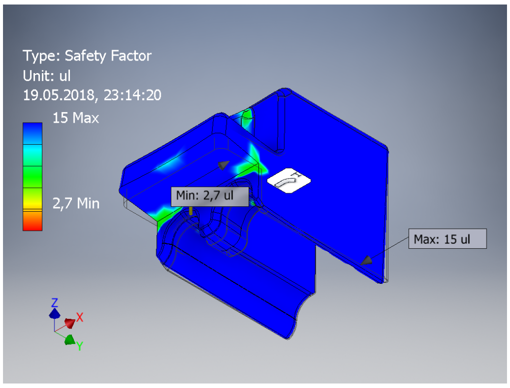


Figure 3.9: Generative design

3.2.5 Design of baseplates

To mount the LEGO plate accurately enough, the 3d printed plate was made, and the LEGO plate was glued on. As a mounting mechanism, tapered legs were used to fit into the holes of the table precisely. Whol baseplate is shown in the picture (PIC), and complete drawings are in the appendix.

For the Shuttles of montrac, the plates with pins corresponding to the size of LEGO pins but with self-accurate mechanism were made by colleague Simon Destrooper. Those plates are in the picture 4.1.

(a) : Bottom side of plate **(b) :** Top side of plate

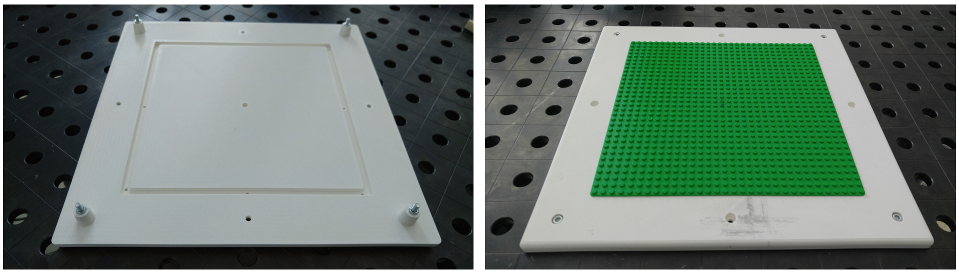


Figure 3.10: Lego base plate.

3.2.6 Skill implementation

The elementary skills for iiwa can be defined as openGripper, closeGripper, moveLin, movePTP, etc. and all of them are described in table 3.3. Those skills are all indivisible in the sense, that they don't have any skills that they can consist of. Superior skills, that consist of those elementary skills and that were used during lego manipulation are pickLego and placeLego. Superior skill to those is buildRing that builds ring from bricks two floors high on given position. Most of these skills have parameters that have to be set. For example skill pickLego have parameters xPosition, yPosition, zPosition, rotation and basePlateNumber and consists of the following operation:

```
movePTP(xPosition, yPosition, zPosition = "10 brick above the origin", rotation,
        basePlateNumber);
openGripper();
moveLin(xPosition, yPosition, zPosition + 2 mm, rotation, basePlateNumber);
closeGripper();
moveLin(xPosition, yPosition, zPosition, rotation, basePlateNumber);
moveLin(xPosition, yPosition, zPosition = "10 brick above the origin", rotation,
       basePlateNumber);
```

A similar approach was used on Agilus robots for building subparts. The final result is, that in LEGO Designer the model of the product consisting of different subparts is made. Planner, in this case, Fast Downward is used to plan assembly of subparts for Agilus robots. Those subparts are then transported to the iiwa working station where iiwa, following the generated plan, assembly subparts into the final product. Assembly itself is done in the noncooperative mode to preserve maximal precision possible. When whole ring or floor was done, iiwa sent the signal to PLC as a confirmation.

Skill	Description	Parameters
openGripper	opens jaws of gripper	-
closeGripper	closes jaws of gripper	-
waitForEdge	robot weits for rising edge from PLC	input
sendSingnal	robot creates rising edge by sending true for one second to PLC	output
moveLin	linear movement to given point	xPosition, yPosition, zPosition, rotation, basePlateNumber
movePTP	movement along the fastest path	xPosition, yPosition, zPosition, rotation, basePlateNumber
moveSaveLin	human safe linear movement to given point	xPosition, yPosition, zPosition, rotation, basePlateNumber
moveSaveLin	human safe movement along the fastest path	xPosition, yPosition, zPosition, rotation, basePlateNumber

Table 3.3: Table of elementary skills.

3.3 Second stage - Connecting ontology database to planner

For this stage, the approach was almost the same as in previous stage, but all assembly was done only on iiwa. Only significant changes made, were the change of source, where LEGO Designer was replaced by ontology and change of planner, because as it turned out, the Fast Downward planner was not able to handle continuous systems. (For Lego assembly was the absence of mathematical operations solved by definition of addOne action, where all possible values had to be listed in initialization. This approach is not feasible in continuous systems, as there is infinite number of possibilities.)

3.3.1 PDDL planner

The goal for the PDDL planner was to plan the sequence of operations to avoid colisions. The information about parts gathered from ontologies should be used to generate problem file for the planner. For the setting of the condition for collision, only simple math such as addition, comparison and subtraction is needed. Those simple mathematicall operations are available in PDDL standard since version 2.1. Another feature that come in handy is Flattening Actions [FL03]. Those actions use quantifiers such as "forall" and "exists" thanks to which it is not necessary to list conditions for all instances.

Another advantage is that it helps us create more general domain definitions so it is not necessary to know the number of instances that will be used in problem definition. Reason for that is that the actions and conditions can be set for all of them generaly. To test this approach, domain definition file was created with the use of features only up to PDDL 2.1. The created domain definition allowed to use multiple baseplates, allowed to compute collision for different rotations of components and allowed to use variable number of components. Unfortunately, the features of numerical planning are not yet supported in Fast Downward planner even when it aims to support PDDL 2.2.

“Fast Downward aims to support PDDL 2.2 level 1 plus the action-costs requirement from PDDL 3.1. For a definition of the various "levels" of PDDL, see p. 63 the paper [FL03].

This means that the following major parts of PDDL are unsupported:

- All aspects of numerical planning. These are introduced at level 2 of PDDL. Exception: some numerical features are part of the: action-costs requirement of PDDL 3.1, and these are supported by the planner.
- All aspects of temporal planning. These are introduced at level 3 of PDDL and above.
- Soft goals and preferences. These are introduced in PDDL 3.0.
- Object fluents. These are introduced in PDDL 3.1. ”[Helb]

Therefore there was the need to find another planner that supports those features. The problem is that there are almost none PDDL planners freely available. Official pages of International Planning Competition contains only papers about algorithms for planning and the links to planners are dead. Through Google, It was possible to find few downloadable planners (most of the links were dead too), but all of them had some limitations. List of tested planners with its limits and problems can be found in the table 3.4. As most promising seemed to be the online planner planning.domains [Mui]. It was able to handle flattening actions and all numerical fluents except subtraction. Using of subtraction was resulting in unsolvable plans even when used as the unary operator or with the different type of dash. Because subtraction is the feature that is indispensable for computing intersections, planner had to be denied. As an alternative was found planner ENHSP [Sca] that is capable of numerical fluents but cannot handle flattening actions, therefore all components have to be listed. For this planner was created "simplified" domain definition for the maximum number of three parts and was successfully tested for different configurations.

In Domain definition file, there are two actions that are based on pick and place skill. In pick-action component is taken and cannot be placed back. After that in preconditions of place-action, collision is checked. If place operation is accident-free, part is placed. The area that is tested for collision is shown in the picture 3.11 where angle can acquire values of 0, 90, 180 or 270 degrees. Corresponding predicates and functions are shown in the following snippet of domain code:

```
(:predicates
            ; Defines position where the planner tries to place component.
            (component_on_place ?comp - component ?pos - position)
            ; Defines which component is held by gripper.
            (gripper_holds_component ?grip - gripper ?comp - component)
            ; The gripper is not used.
            (gripper_free ?grip - gripper)
            ; The component is on its starting position.
            (component_free ?comp - component)
            ; The position is not occupied by a component.
            (position_free ?pos - position)
            ; Checks if two given positions are the same objects.
            (equals ?pos1 ?pos2 - position)
            ; True if object should be part of plan (enambles general solution for max 3 objects)
            (position_is_used ?pos - position)
            ; True if object should be part of plan (enambles general solution for max 3 objects)
            (component_is_used ?comp - component)
    )

    (:functions
        ; X coordinate of the final position of the component [mm]
        (position_x_coordinate ?pos - position)
        ; Y coordinate of the final position of the component [mm]
        (position_y_coordinate ?pos - position)
        ; Z coordinate of the final position of the component [mm]
        (position_z_coordinate ?pos - position)
        ; Angle of the final rotation of the component [deg]
        (position_rotation ?pos - position)
        ; X coordinate of the position of the component in a storage [mm]
        (component_x_coordinate ?comp - component)
        ; Y coordinate of the position of the component in a storage [mm]
        (component_y_coordinate ?comp - component)
        ; Z coordinate of the position of the component in a storage [mm]
        (component_z_coordinate ?comp - component)
        ; Angle of the rotation of the component in a storage [deg]
        (component_rotation ?comp - component)
        ; Width of the component's body divided by two [mm] (distance between open fingers)
        (component_half_width ?comp - component)
        ; Distance of the center of the component body from the final position along x axis [mm]
        (component_x_displacement ?comp - component)
        ; Distance of the center of the component body from the final position along y axis [mm]
        (component_y_displacement ?comp - component)
        ; Width of the gripper's finger [mm]
        (finger_width ?grip - gripper)
        ; Length of the gripper's finger [mm]
        (finger_half_length ?grip - gripper)
    )
```

3.3.2 3D printing of board and gripper fingers

For the circuit component assembly, new fingers were printed based on experience with Lego manipulation task. The fingers for circuit assembly are modified version of the third version of Lego gripper. As there was no need

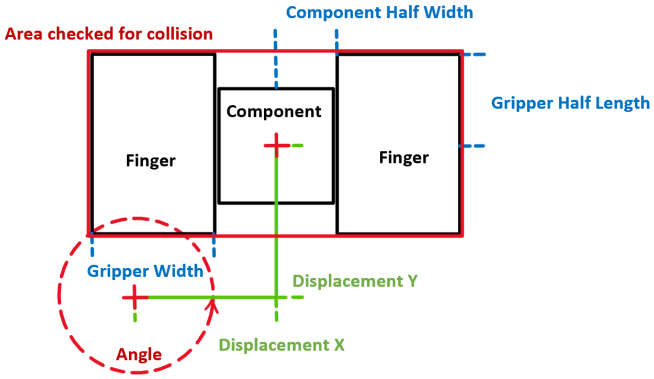


Figure 3.11: Checking for collision in PDDL action

for hard pressing of electronic components the protrusion and the tooth was removed. Also, the finger had to be done a little bit thicker because the parts chosen for assembly are narrower. The fingers are shown in the picture (PIC). The drawings are in the appendix.

To provide maximum accuracy during the assembly process, the circuit board holder was made. This holder was designed to fit the sizes of the particular board to anchor its position. For the simplification of the task, the holder was designed so it can fit on Lego base plate. Thanks to this feature all bases, calibrated for Lego task, could be used without change.

For the store, the used approach was similar. The component holder was made to fit Lego plates too, and the design was made to fit particular components. The whole setup is shown in the picture 4.1.

3.3.3 Robot moves

For this task, the movements were first set to the noncooperative mode to minimize the inaccuracy and to test the repeatability of the assembly with given components as the legs of the components were loose. After the calibration of the position of individual parts, the test was run. During ten reps there was no component wrongly placed.

After the successful test, it was decided to switch robot into the cooperative mode. To maintain the accuracy, the speed of the robot was reduced, and the joints were set stiff, so there was minimal spring effect. The robot was tuned, so it stops when the external force acting on the end effector reaches about 2 newtons of static force.

3.3.4 Ontology

Rosetta otology [Mal] strongly influenced the design of ontologies mentioned in diagram 3.1. Rosetta ontology cannot be used directly because it suffered from some missing or different properties, and classes that were needed for this approach. To make things simpler, only required parts of Rosetta ontology were taken, but following the same layout where it was possible. The main ontology tree can be seen in the picture 3.12.

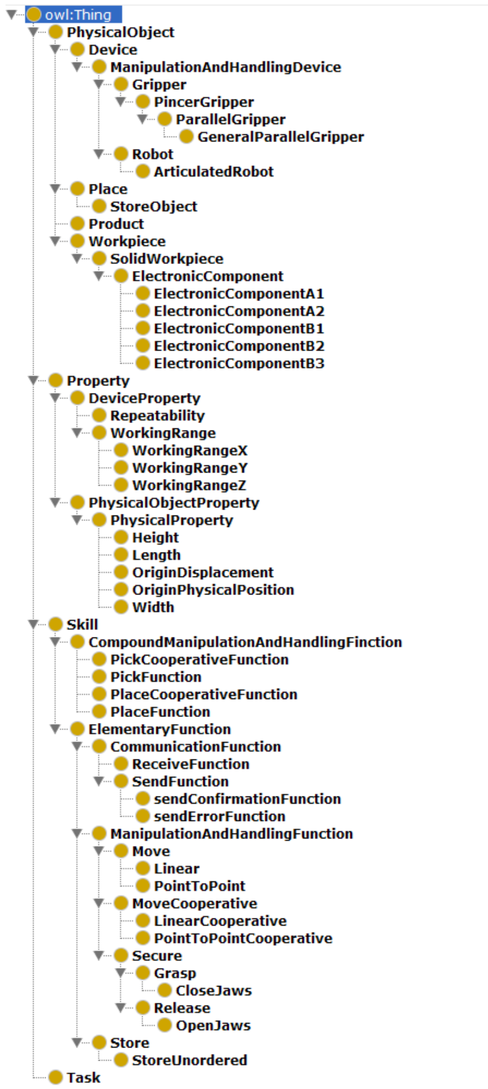


Figure 3.12: Classes of used ontology shown in form of tree

The main ontology is the description of capabilities of the robotic unit. This ontology consists of different devices where each device has listed all available skills through inverse hasSkill/isSkillOf property.

The Store ontology is the description of all parts available. Those parts are linked to the storeUnsorted skill through inverse hasComponent/isComponent property. The skill itself is linked through property hasSkill/isSkillOf.

Finally, the product ontology consists of product class, which has connected to it the list of needed parts and task that contains all skills needed. All components stored, contains data properties with dimensions and measurements. Those data are then used in planner.

Described ontologies are used for comparison capabilities of robotic unit with requirements of product. The comparison itself is separated into two phases, wherein the first phase, the main ontology is compared with product ontology, to see if all skills are available. In the second phase, the product ontology is compared to ensure component availability.

For this comparison, the SPARQL and reasoner are needed [JOM17]. Because the comparison would be difficult, if not impossible, only with the use of SPARQL and reasoner, java library Apache Jena was used, to loop over the list of all skills or components. Apache Jena was used instead of OwlApi, asi it has better documentation. The used queries are shown in the following example.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX def:<http://www.ciirc.cvut.cz/default#>
SELECT ?skill

where {
    def:ProductA def:hasTask ?t .
    ?t def:hasSkill ?skill
    }
```

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX def:<http://www.ciirc.cvut.cz/default#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT distinct ?device

where {
        < skill.asResource().getURI()> def:hasSubskill+ ?elementarySkill .
         ?device def:hasSkill ?elementarySkill .
         ?device rdf:type ?deviceConcept .
         ?deviceConcept rdfs:subClassOf+ def:Device .
         }
```

The first query is used to find all skills in the product. The second one is used in a loop to check if the found skill is elementary skill and then compared to those available in unit. In case of the store is the situation similar.

Library	Planner	Platform	Numerical fluents	Flattening actions	Others
PDDL4J	FF	Java	No	No	-
	HSP	Java	No	No	-
	EHC	Java	No	No	-
planning.domains	?	Web	Yes	Yes	Bug when subtracting
DiNo *	SRPG+	C++ (Linux)	?	?	unable to run
ENHSP	gbfs+hadd	Java 1.8	Yes	No	-
	wa_star+hadd+hw	Java 1.8	Yes	No	-
	wa_star+hadd+hw**	Java 1.8	Yes	No	-
	a_star+hrma	Java 1.8	Yes	No	-
	a_star+aibr h	Java 1.8	Yes	No	-
	a_star+hlm	Java 1.8	?	?	needs cplex 12.6.3

Table 3.4: Summary of found PDDL planners: PDDL4J [Pel], DiNo [Lon], planning.domains [Mui], ENHSP [Sca]

Chapter 4

Tests

To test the setup, five checks were made with use of two different circuit boards. The first two tests were made to try out assembly without collision. During those tests, only two components were used, in each case with different placement in the store. After that two tests were made to proof collision detection during planning, so a sequence of operations was chosen accordingly. The last test was designed, so there is no sequence how to solve it.

The setup of those tests is in the pictures 4.2, 4.3. Values of the planner as well as planning time and plan length are in the table 4.1. All of those tests were made with the greedy best-first search algorithm. To see the performance of all available algorithms task 1B was planned with the use of each algorithm. The comparison is in the table 4.2. To know repeatability of circuit assembly all four solvable tasks were performed ten times in a row. Except for test A1, all tasks had 100% success rate. During the test A1 one component was inaccurately placed.

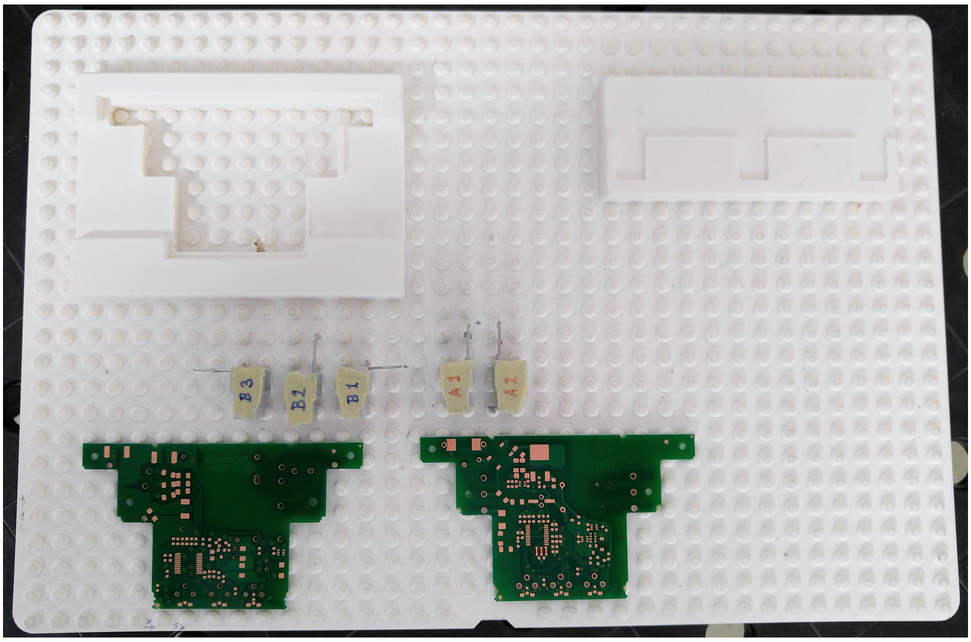


Figure 4.1: The setup used for tests of assembly

(a) : Test 1A **(b) :** Test 2A **(c) :** Result of 1A, 2A

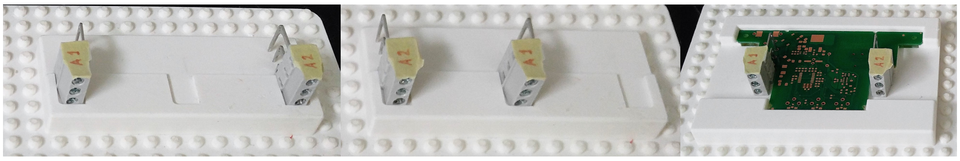


Figure 4.2: First two tests and wanted result

(a) : Test 1B **(b) :** Test 2B **(c) :** Result of 1B, 2B

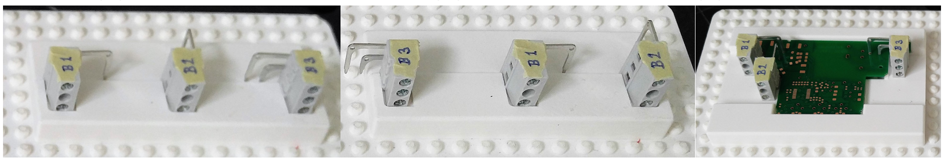


Figure 4.3: Second two tests and wanted result

	1A	2A	1B	2B	1C
Plan Length	4	4	6	6	-
Planning Time [ms]	700	691	862	939	1015
Heuristic Time [ms]	2	5	31	10	21
Search Timel [ms]	144	119	232	231	0
Expanded Nodes	5	5	12	9	9
States Evaluated	7	7	23	19	32

Table 4.1: Values of planner during tests with use of greedy best-first search. Test 1C is test with 3 parts that collide with each other

Algorithm	P len	P time	H time	S time	Expand	States
gbfs+hadd	6	880	17	288	12	26
wa-star+hadd+hw	6	859	20	274	12	26
wa-star+hadd+hw	6	847	31	264	7	15
a-star+hrmax	6	877	20	235	13	31
a-star+aibr	6	705	23	143	7	15

Table 4.2: Test of different planning algorithms on task 1B.Time is measured in ms. P len - Plan Length, P time - Planning time, H time - Heuristic time, S time - Search time, Expand - Expanded nodes, States - States evaluated

Chapter 5

Conclusion

In this thesis, the knowledge-based approach for printed circuits assembly was presented. The decisionmaking architecture for this approach includes ontologies, reasoner and SPARQL, and planner. Thanks to the use of planner, there is no need for storing a sequence of operations, needed for manufacturing, in the product description. To implement and experimentally verified suggested approach, supportive tools, such as gripper fingers and circuit board holders were designed and tested with use of finite element stress analysis. For optimization of finger model, generative design tool, which uses AI for model optimization, was used and compared with the manually designed model. For accurate cooperative robot movements, calibration methods were suggested and implemented.

For this approach, ontologies for the product, robotic unit, and store were created based on ROSETTA ontology. Those ontologies are describing capabilities of devices with use of "skills." Availability of required skills is evaluated with use of reasoner and SPARQL. For the planning part of the solution, the research of available planners was made, and the best-suited option was chosen. Then planning domain and planning problem were created to meet the needs of the circuit assembly. Finally, the implemented solution was successfully experimentally tested on the assembly of two different circuit boards.

5.1 Future work

For improvement of the implemented solution, there are many things that can be done. One of biggest improvements due to the current state would be development or acquisition of PDDL planner, that would support PDDL 3.1 standard. In future, the planner can be used to handle unpredictable events such as device failure. Another significant improvement of existing solution would be a generalization on arbitrary assembly task and automatization of approach.

Appendix A

Bibliography

[AG] SICK AG, *Safety laser scanners s3000 profinet io advanced*, https://www.sick.com/cz/en/opto-electronic-protective-devices/safety-laser-scanners/s3000-profinet-io-advanced/s30a-4111cp/p/p120140, Accesed: 2018-05-11.

[Ame] PI North America, *Profinet, industrial ethernet for advanced manufacturing*, https://us.profinet.com/technology/profinet/, Accesed: 2018-05-11.

[BBM10] L. Belouaer, M. Bouzid, and A. I. Mouaddib, *Ontology based spatial planning for human-robot interaction*, 2010 17th International Symposium on Temporal Representation and Reasoning, Sept 2010, pp. 103–110.

[BFKR14] Malte Brettel, Niklas Friederichsen, Michael Keller, and Marius Rosenberg, *How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective*, International Journal of Mechanical, Industrial Science and Engineering **8** (2014), no. 1, 37–44.

[CCGPN17] J.M. Chacón, M.A. Caminero, E. García-Plaza, and P.J. Núñez, *Additive manufacturing of pla structures using fused deposition modelling: Effect of process parameters on mechanical properties and their optimal selection*, Materials & Design **124** (2017), 143 – 157.

[Con] Maurice Conti, *The incredible inventions of intuitive ai*, https://www.ted.com/talks/maurice_conti_the_incredible_inventions_of_intuitive_ai, Accesed: 2018-05-20.

[DECA18] John Ryan C. Dizon, Alejandro H. Espera, Qiyi Chen, and Rigoberto C. Advincula, *Mechanical characterization of 3d-printed polymers*, Additive Manufacturing **20** (2018), 44 – 67.

[FES] FESTO, *Bioniccobot*, https://www.festo.com/group/en/cms/12746.htm, Accesed: 2018-05-11.

[FES17] ———, *Parallel grippers dhps*, October 2017.

[FL03] Maria Fox and Derek Long, *Pddl2. 1: An extension to pddl for expressing temporal planning domains*, Journal of artificial intelligence research (2003).

[GL05] Alfonso Gerevini and Derek Long, *Plan constraints and preferences in pddl3*, Tech. report, Technical Report 2005-08-07, Department of Electronics for Automation, University of Brescia, Brescia, Italy, 2005.

[Gmba] KUKA Roboter GmbH, *Kr agilus*, https://www.kuka.com/en-at/products/robotics-systems/industrial-robots/kr-agilus, Accesed: 2018-05-10.

[Gmbb] ———, *Lbr iiwa*, https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa, Accesed: 2018-05-11.

[Gmbc] Montratec GmbH, *Montrac*, http://www.montratec.de/en/, Accesed: 2018-05-10.

[Gmb15] KUKA Roboter GmbH, *Robots kr agilus sixx with w and c variants - operating instruction*, March 2015, Version: BA KR AGILUS sixx V12.

[Gmb16] ———, *Kuka sunrise.os 1.11, kuka sunrise.workbench 1.11 - operating and programming inst ructions for system integrators*, July 2016, Accesed: 2018-05-11.

[Groa] OWL Working Group, *Owl 2 web ontology language - document overview (second edition)*, https://www.w3.org/TR/owl2-overview/, Accesed: 2018-05-20.

[Grob] ———, *Owl web ontology language semantics and abstract syntax*, https://www.w3.org/TR/owl-semantics/syntax.html, Accesed: 2018-05-20.

[Groc] SCHMID Group, *Module line: Shuttles instead of conveyors*, https://schmid-group.com/en/schmid-group/news-events/press-releases/module-line-shuttles-instead-of-conveyors, Accessed: 2018-05-20.

[Grod] Stanford Robotics Research Group, *Fast research interface library - manual and documentation*, https://cs.stanford.edu/people/tkr/fri/html/, Accesed: 2018-05-12.

[Groe] W3C RDF Data Access Working Group, *Sparql query language for rdf*, https://www.w3.org/TR/rdf-sparql-query/, Accesed: 2018-05-20.

[Hela] Malte Helmert, *Pddl papers*, http://icaps-conference.org/ipc2008/deterministic/PddlResources.html, Accesed: 2018-05-11.

[Helb] ______, *Pddl support*, http://www.fast-downward.org/PddlSupport, Accesed: 2018-05-20.

[HUB] 3D HUBS, *Additive manufacturing technologies: An overview*, https://www.3dhubs.com/knowledge-base/additive-manufacturing-technologies-overview#/binder-jetting, Accesed: 2018-05-11.

[Inc] Autodesk Inc, *1st principal stress*, https://knowledge.autodesk.com/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Inventor-Help/files/GUID-8C3D24BC-D259-40D6-9543-6032B1B9B360-htm.html, Accesed: 2018-05-20.

[Jac15] Ludwig Jacobsson, *A module-based skill ontology for industrial robots* (eng), Student Paper.

[Job] Jobspin, *Czech job market: Trends in 2017*, https://www.jobspin.cz/2017/01/czech-job-market-trends-in-2017/, Accesed: 2018-05-11.

[JOM17] V. Jirkovský, M. Obitko, and V. Mařík, *Understanding data heterogeneity in the context of cyber-physical systems integration*, IEEE Transactions on Industrial Informatics **13** (2017), no. 2, 660–667.

[Lon] King's College London, *Dino*, https://github.com/KCL-Planning/DiNo, Accesed: 2018-05-23.

[Mal] Jacek Malec, *The rosetta ontology of industrial robotics devices and robotic skills*, https://github.com/jacekmalec/Rosetta_ontology, Accesed: 2018-05-23.

[Man] ManpowerGroup, *Czech employers report cautiously optimistic hiring plans for 4q 2016*, https://www.manpower.cz/manpower/wp-content/uploads/2016/09/Manpower-Employment-Outlook-Survey-q4-2016-press-release.doc, Accesed: 2018-05-11.

[MAR15] Muhayyuddin, A. Akbari, and J. Rosell, *Ontological physics-based motion planning for manipulation*, 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), Sept 2015, pp. 1–7.

[MNB13] Jacek Malec, Klas Nilsson, and Herman Bruyninckx, *Describing assembly tasks in declarative way - extended abstract*, MNB WS ICRA (2013).

[MPB+11] E. Miguelanez, P. Patron, K. E. Brown, Y. R. Petillot, and D. M. Lane, *Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles*, IEEE Transactions on Knowledge and Data Engineering **23** (2011), no. 5, 759–773.

[Mui] Christian Muise, *Pddl support*, `http://editor.planning.domains/`, Accesed: 2018-05-20.

[MW] Incorporated Merriam-Webster, *factor of safety*, `https://www.merriam-webster.com/dictionary/factor%20of%20safety`, Accesed: 2018-05-20.

[Nie96] Ilkka Niemelä, *A tableau calculus for minimal model reasoning*, International Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Springer, 1996, pp. 278–294.

[Pel] Damien Pellier, *pddl4j*, `https://github.com/pellierd/pddl4j`, Accesed: 2018-05-23.

[PHK14] Mikkel Rath Pedersen, Dennis Levin Herzog, and Volker Krüger, *Intuitive skill-level programming of industrial handling tasks on a mobile manipulator*, Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 4523–4530.

[Rob] ABB Robotika, *Yumi® - creating an automated future together.you and me.*, `http://new.abb.com/products/robotics/industrial-robots/yumi`, Accesed: 2018-05-12.

[Sca] Enrico Scala, *The enhsp planner*, `https://gitlab.com/enricos83/ENHSP-Public`, Accesed: 2018-05-23.

[Sim] SimScale, *Fea*, `https://www.simscale.com/docs/content/simwiki/fea/what-is-von-mises-stress.html`, Accesed: 2018-05-20.

[SK08] Bruno Siciliano and Oussama Khatib, *Springer handbook of robotics*, Springer Science & Business Media, 2008.

[Ste11] Maj Stenmark, *Integration of semantic knowledge to enable re-use of robot programs*, Citeseer, 2011.

[Ste15] ———, *Instructing industrial robots using high-level task descriptions*, Ph.D. thesis, Departement of Computer Science, Lund University, 2015.

[TBFA05] S. Thrun, W. Burgard, D. Fox, and R.C. Arkin, *Probabilistic robotics*, Intelligent robotics and autonomous agents, MIT Press, 2005.

[Vir] Salvo Virga, *cabinet setup*, `https://github.com/IFL-CAMP/iiwa_stack/wiki/cabinet_setup`, Accesed: 2018-05-12.

[WSOG15] Stephan Weyer, Mathias Schmitt, Moritz Ohmer, and Dominic Gorecky, *Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems*, Ifac-Papersonline **48** (2015), no. 3, 579–584.

Appendix B

Drawings

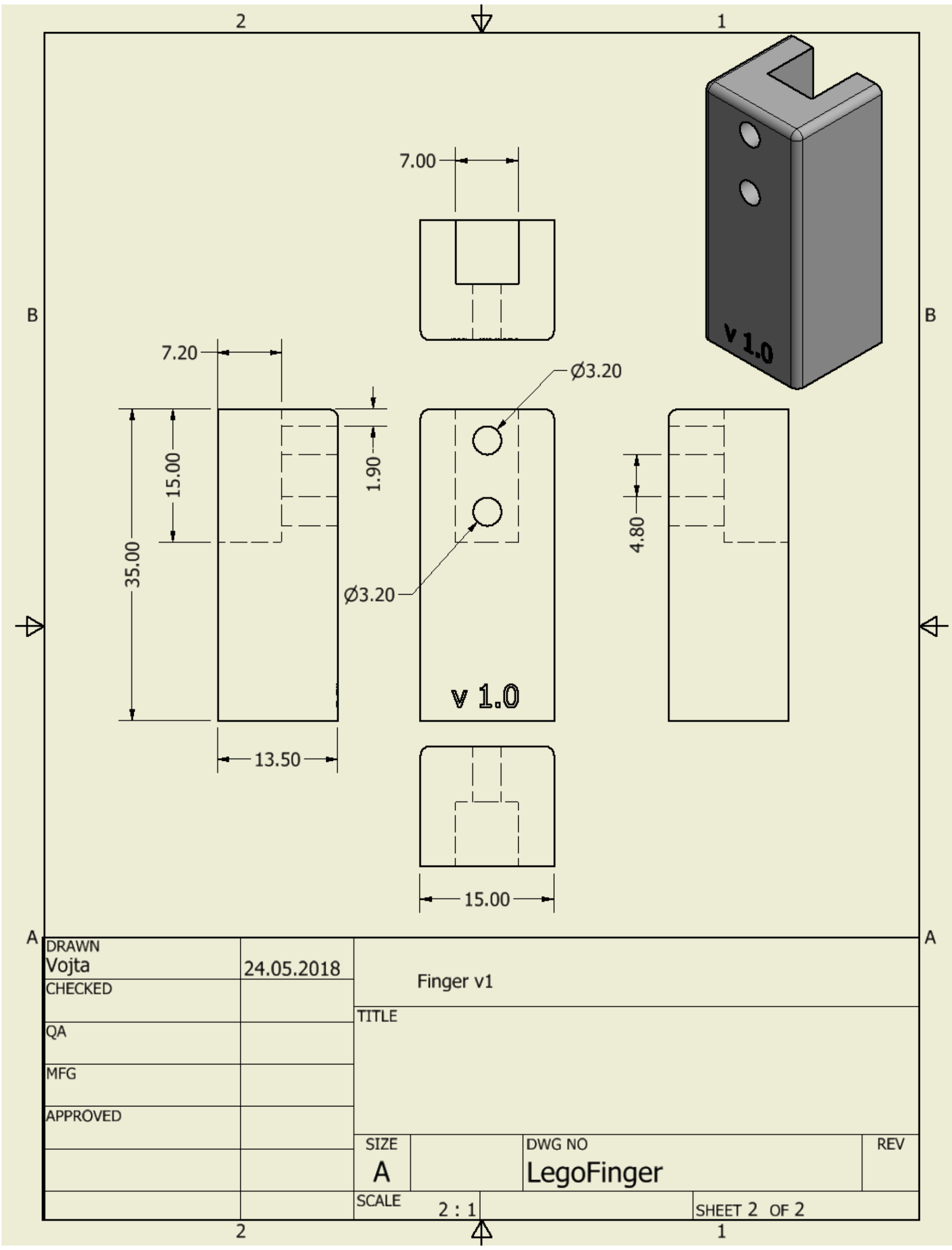


Figure B.1: First version of finger for manipulation of LEGO

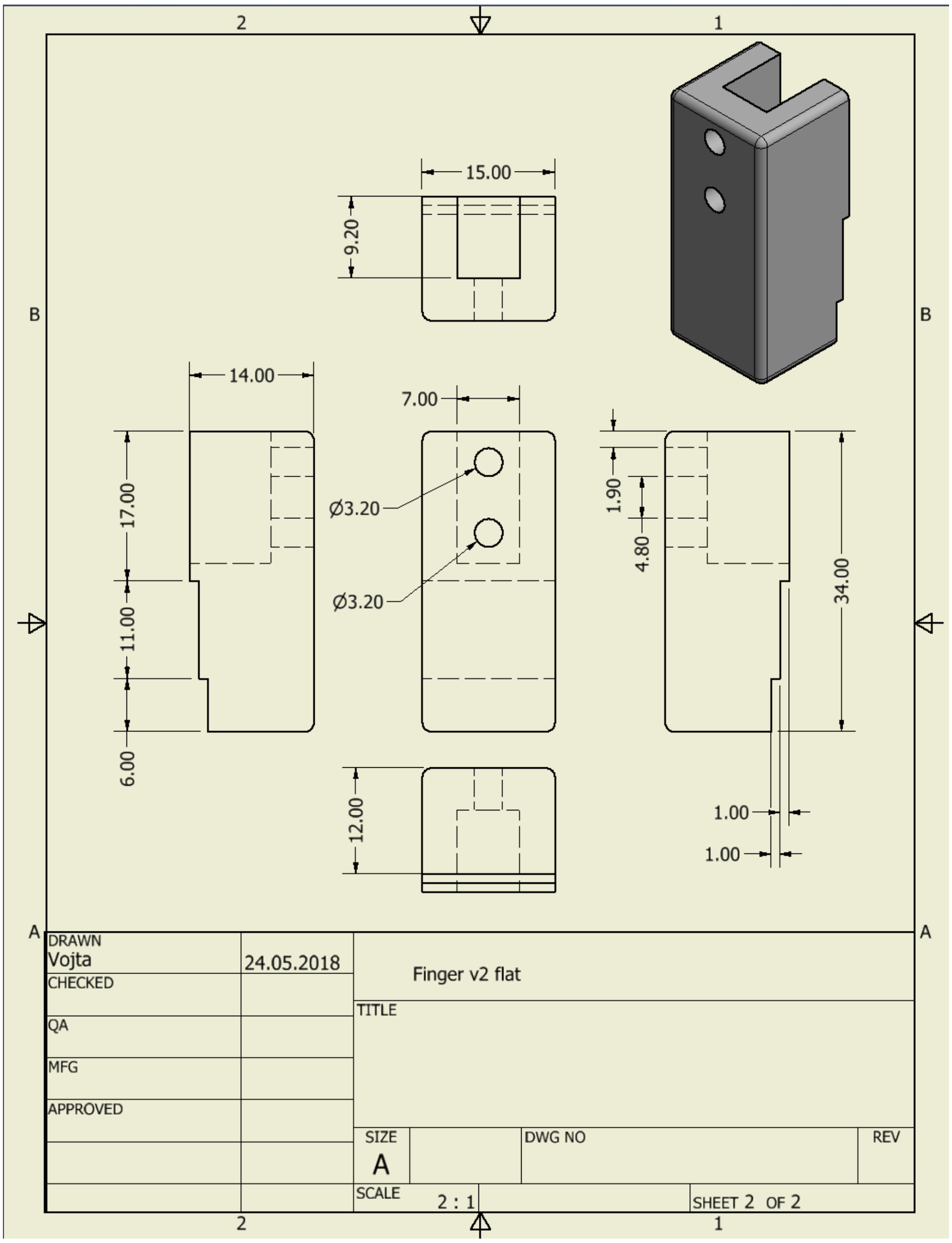


Figure B.2: First version of finger for manipulation of LEGO

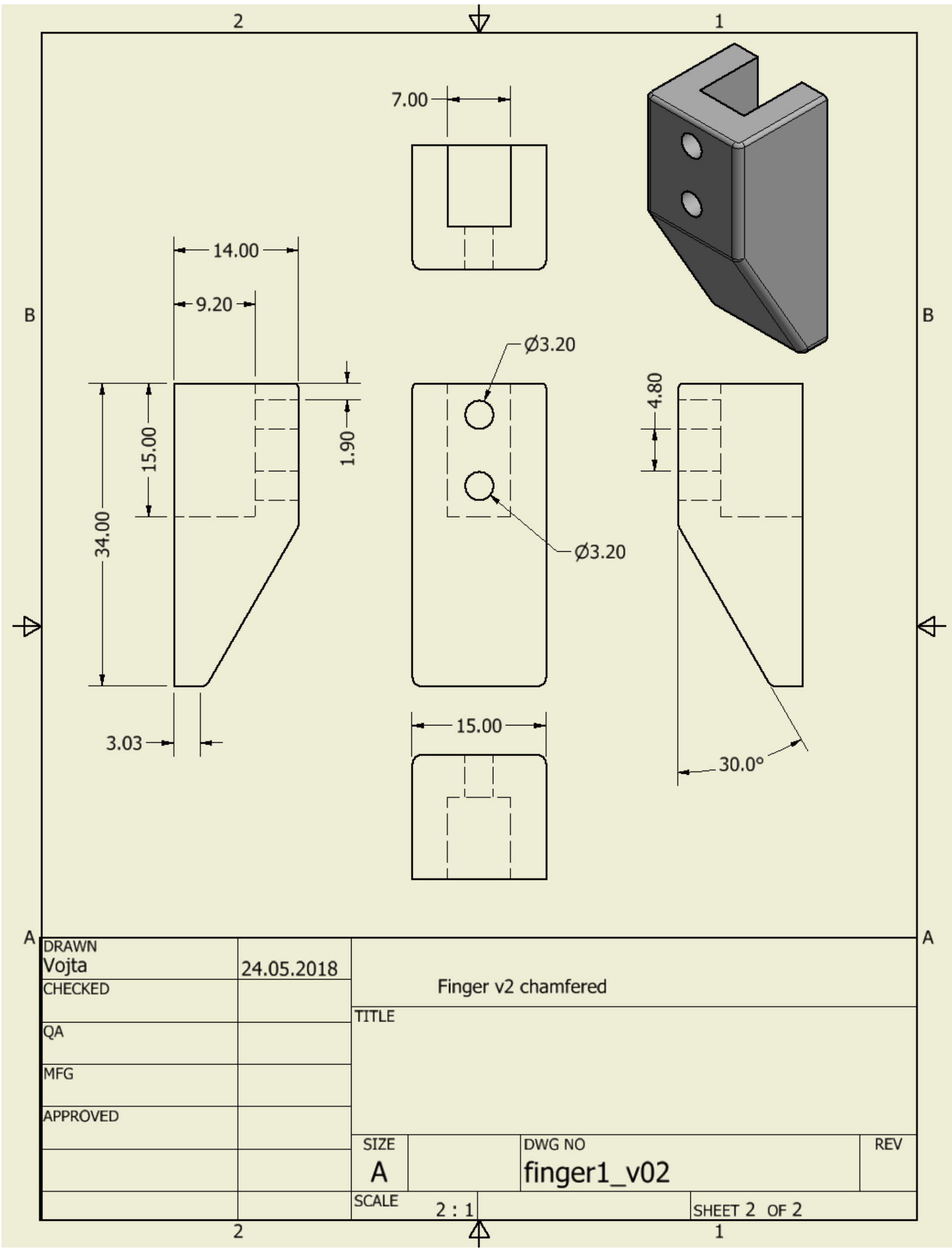


Figure B.3: First version of finger for manipulation of LEGO

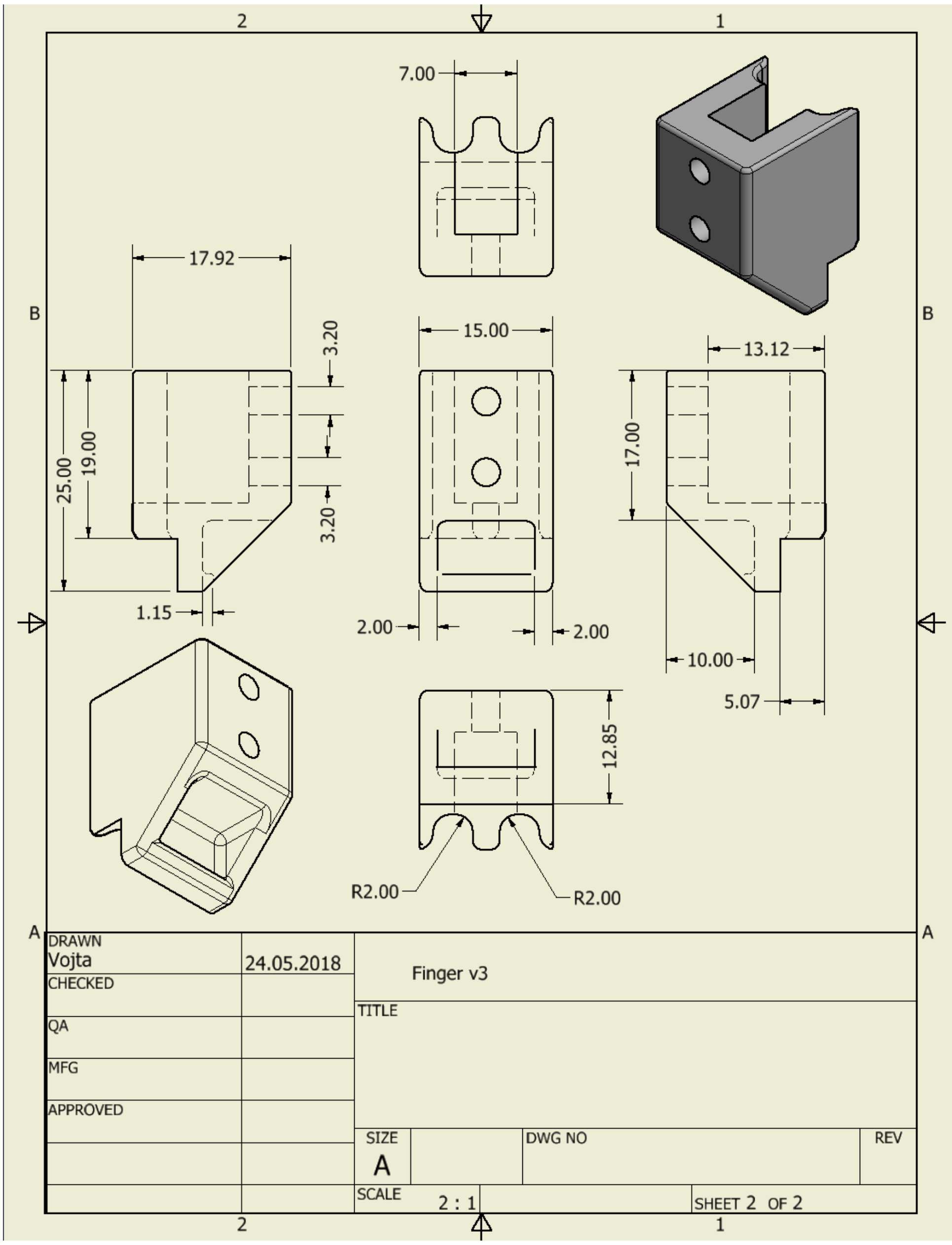


Figure B.4: First version of finger for manipulation of LEGO

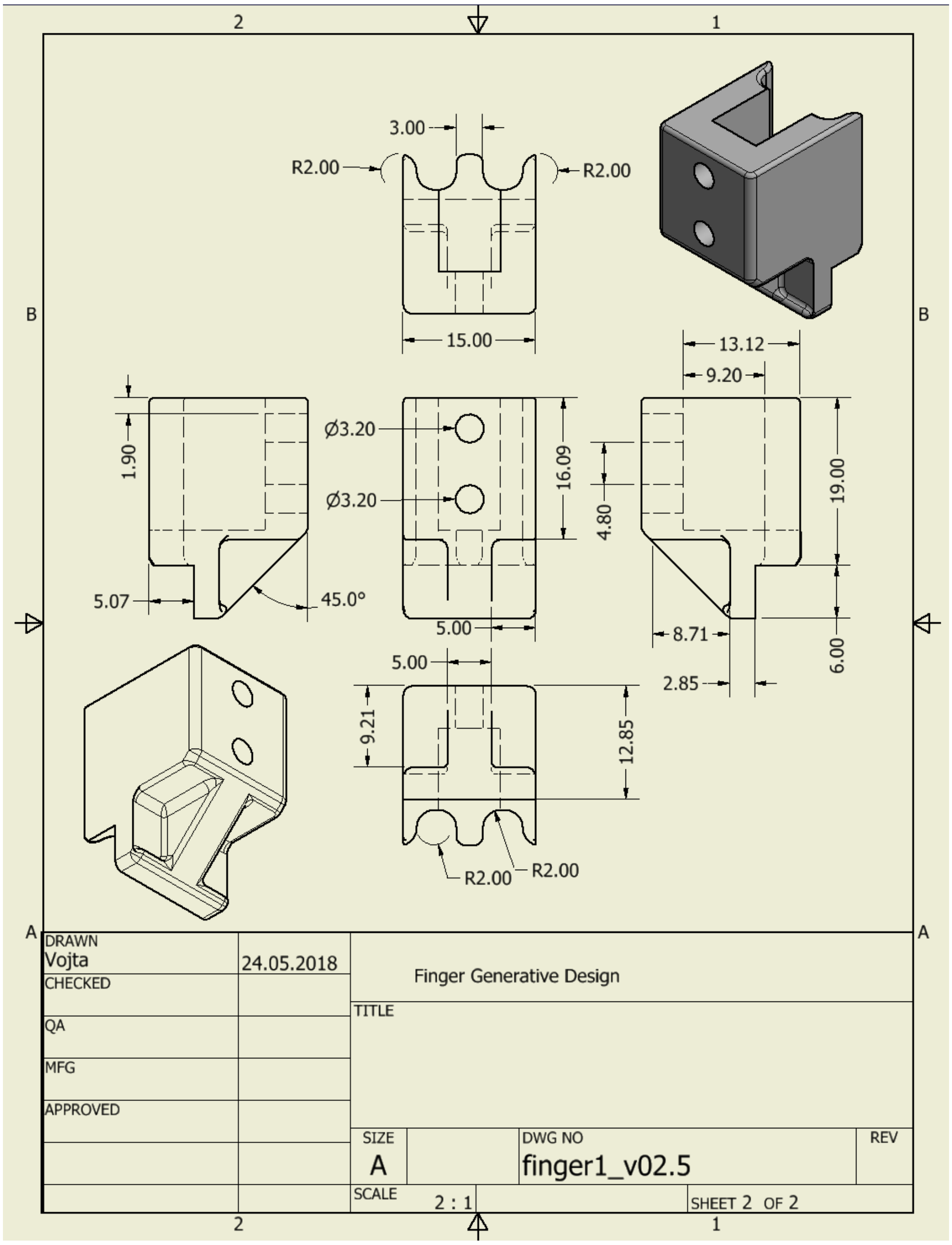


Figure B.5: First version of finger for manipulation of LEGO

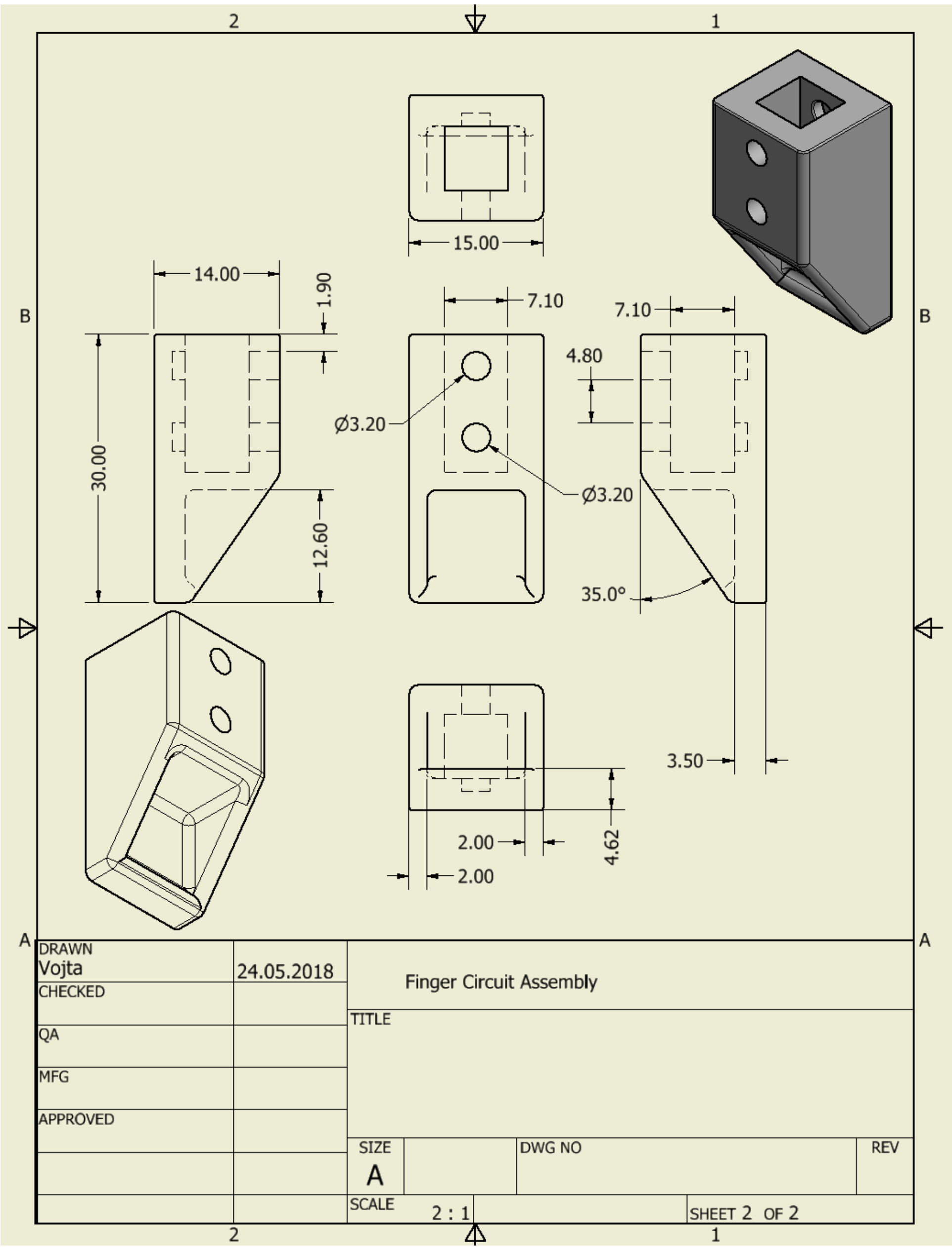


Figure B.6: First version of finger for manipulation of LEGO

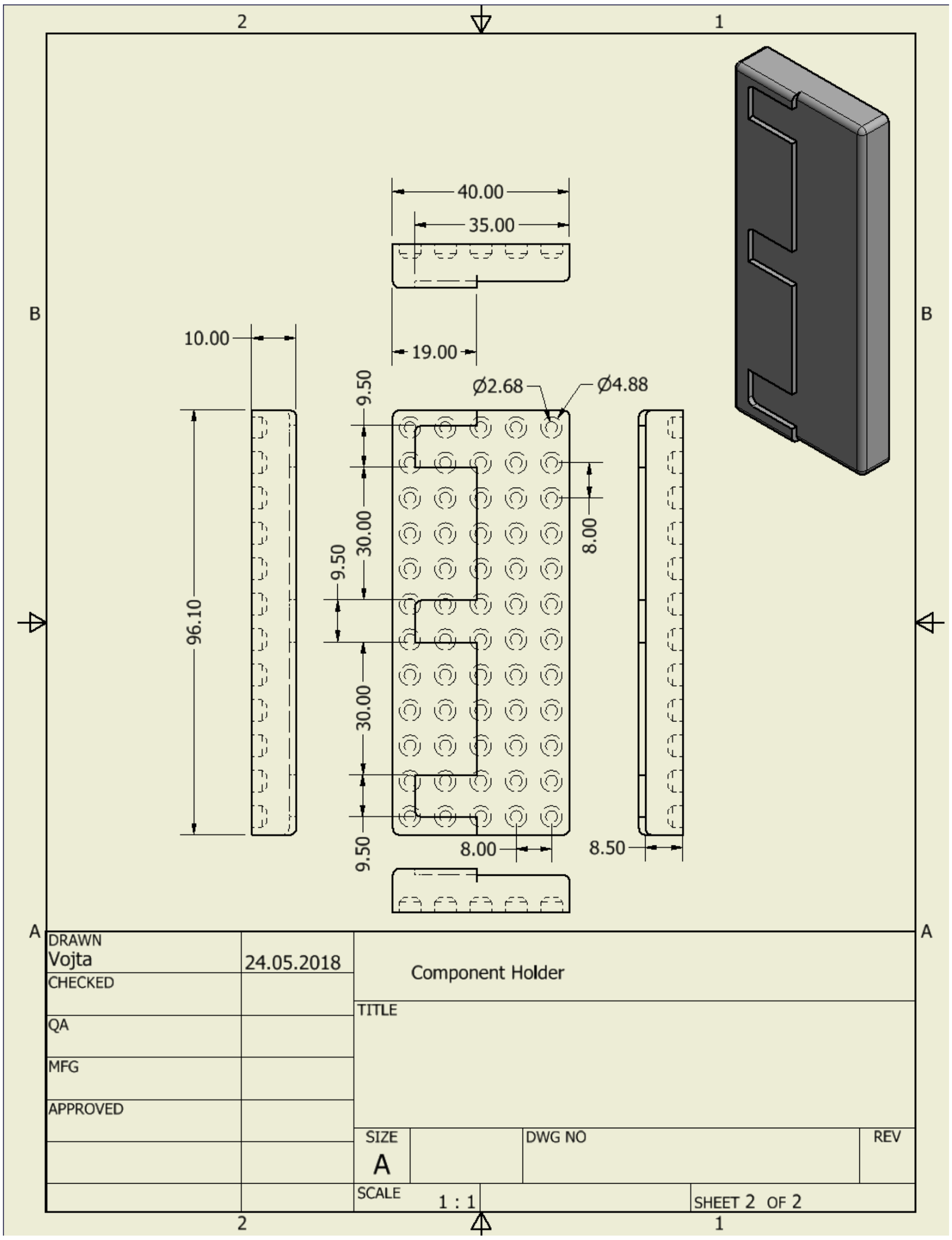


Figure B.7: First version of finger for manipulation of LEGO

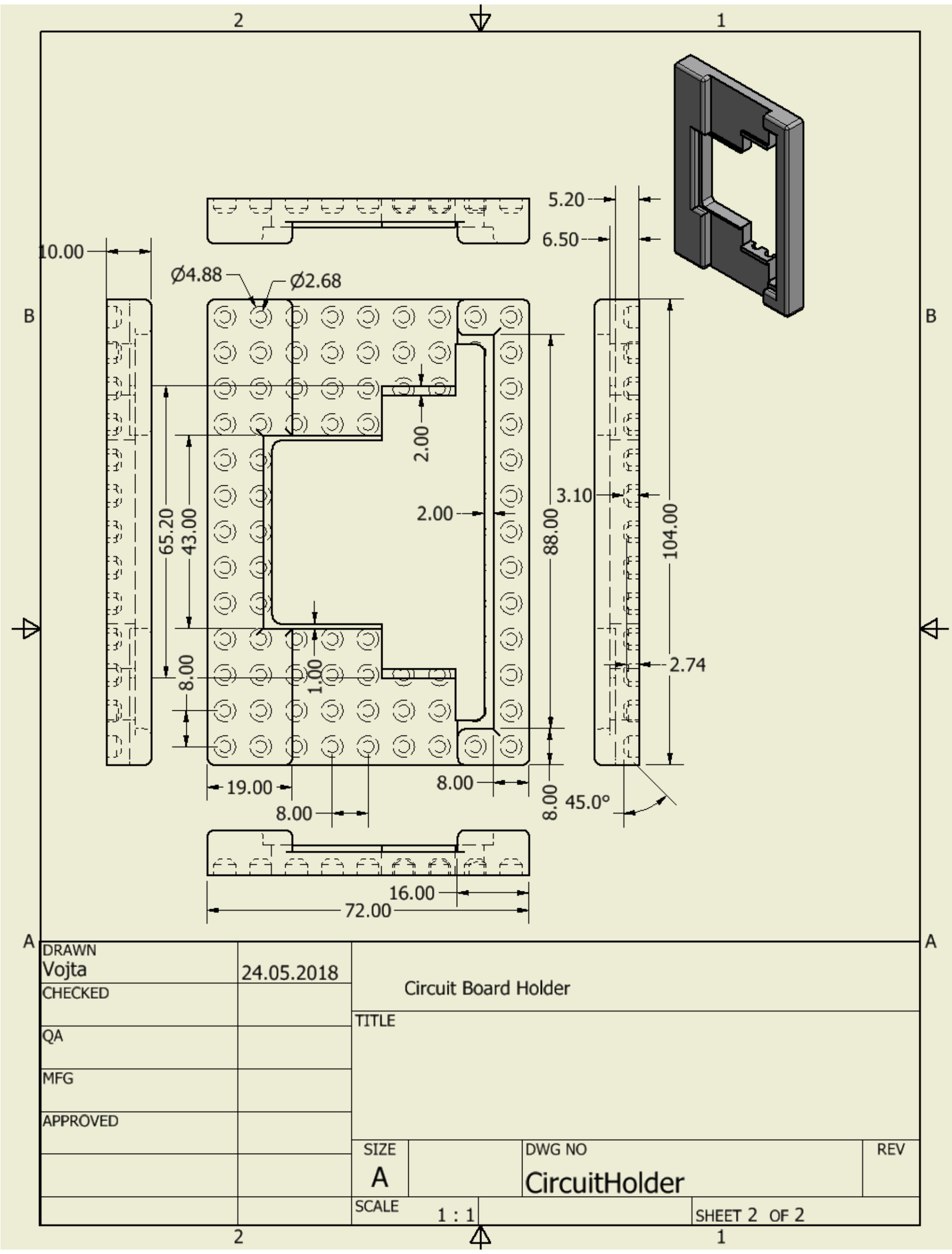


Figure B.8: First version of finger for manipulation of LEGO