ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

BIOLOGICALLY INSPIRED
ROBOTICS GROUP (BIRG)

Summer Internship 2004:

Modular Robot Unit – Characterisation, Design and Realisation

Elmar Dittrich,
Department of Mechanical Engineering,
Technical University Ilmenau,
Germany.

Supervisor: Prof. Auke Jan Ijspeert
BIRG, Logic Systems Laboratory,
School of Computer and Communication Sciences,
Swiss Federal Institute of Technology Lausanne.

16th August 2004

Contents

List of Figures

List of Tables

1. Introduction

A modular robot can be defined as a robotic system consisting of a set of discrete components (or LEGO-like building blocks). This special field of robotics holds many interesting challenges, not only in terms of mechanical engineering, especially in the field of mechatronics (MEMS[1], smart actuators) as well as in terms of control strategies (distributed control, autonomous strategies). There is also a parallel to biological systems: Built from only 100 different cell types, biological systems represent thousands of different organisms and functionalities.

1.1. Motivation and Goals

Although the existing modular robots are still far away from any real application there are interesting scenarios in which the robot could operate. Imagine a building that has been damaged by earthquake, containing a variety of obstructions and possibly not being suitable for any particular standard robot. A self-reconfigurable modular robot could adopt to the current environment. Assuming it has enough sensory input and some algorithm that would allow to change the shape immediately, it could crawl through narrow holes. In the inside of the building it could reshape into a legged robot and walk over rubble, climb stairs or even reach surfaces much higher than itself.

With our first prototype at BIRG (**B**iology **I**nspired **R**obotics **G**roup) we follow up simpler approach. By using mostly off-the-shelf technology, combined with a low-end, but robust design we make the construction cheap and easy to manufacture. These constraints result in the possibility to manufacture a huge number of pieces. State of the art devices are used for communication (Bluetooth) and control (**F**ield **P**rogrammable **G**ate **A**rray). Therefore, the circuit boards are already in development by another student in the lab but should be integrated in the existing prototype.

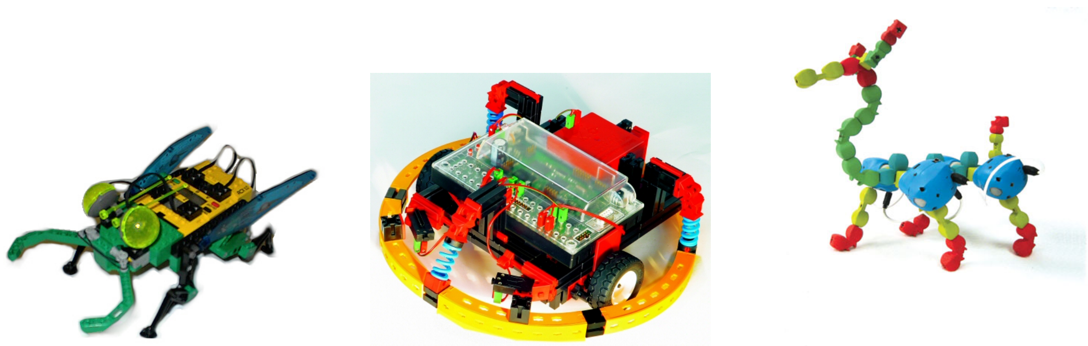
After more then one simulations, concerning self-organisation of locomotion with help of evolutionary algorithms were done at the lab, it was time to design some physical

[1] abbr.: Micro-Electrical-Mechanical-Systems

hardware. One can study multiple kinds of locomotion with the existing prototype. When the communication will be integrated, aspects of distributed control strategies and communication technologies can also be examined. The advantages of modular robotics compared to conventional robots are enormous: it increases the types of robots you can simulate, like snake-robots, several legged-robots and any arbitrary structure. This includes the field of collective robotics and inter-robot behavior as well.

1.2. General Properties of Modular Robots

When looking at all the projects concerning modular robotics, even more simulations then real physical solutions exist, due to the problems of manufacture and control of the simulated systems. The very first ideas came from Von Neumann's self-reproducing automata from the 1960's as the origin of cellular automata models [32]. Nevertheless, it is very difficult to construct these models as real hardware. The next category can be described as statically configured with the help of an operator. This reminds on the famous robotic-invention-systems like LEGO Mindstorms® and Fischertechnik's ROBO Mobile Set. A very recent application comes from the MIT Media Lab and is called Topobo [21]. The biggest effort on the design of modular robots is put in self-reconfigurable structures, sometimes called metamorphic. This means that the modules can change their position and thus their connections automatically without manual help. Self-reconfiguration also opens the possibility of self-repair.



(a) (b) (c)

Figure 1.1.: LEGO Mindstorm® (a), Fischertechnik (b) and Topobo (c)

Homogeneity vs. Heterogeneity Homogeneity of the modules means that all the elements of the system are identical. In a robot made of homogeneous reconfigurable units, however, each unit can be replaced by any other unit by changing the connection. This means, that the *position* of the module in the robot defines its function. It could act as a head, leg or spine depending on its location in the robot. Homogeneity also has an advantage concerning self-repair, because it need not be known which kind of module failed. At last, the production cost could be lowered due to mass production, simple inspection and maintenance procedures.

If a modular structure consists of several, non-identical parts, it is called heterogeneous. Here the *function* of the modules defines its position in the robot. Although for "real" modularity it is essential to have identical elements, it is very difficult to transform this into physical elements. Because in homogeneous structures every module must have its own CPU, power supply, sensors and communication interfaces, it will be physically large.

Chain- vs. Lattice-Structure There is another classification in which modular robots can be separated. Chain-like structures remind on biology inspired robots like snakes, spiders, centipedes. In this configuration, the modules are arranged in sequence but can also branch off if desired. Although the chain-like robots are able to move in different gaits or even roll, forming a wheel, they are mostly controlled centralized according to gait-control tables.

One can compare a lattice structure like a grid on a checkerboard, where every square defines a position of an element. Locomotion is provided by reconfiguring its arrangement, thus moving its center of mass of the whole structure. It can be described more like a flow of liquid. Of course these modules have to be homogeneous, because they only have to know its neighbors and theoretically there are no obstacles that can not be overcome, assumed that there are enough modules available. Lattice configuration offers one important advantage: Because positions are limited to the discrete grid space, they do not have to monitor all their cells to avoid different parts smacking into one another. But for todays applications the locomotion of such architecture is too slow and far away from flowing like water.

Distributed Control The problem of decentralized control in distributed robots (even every dynamic reconfigurable network, like swarm robotic systems and distributed sensor networks) can be divided in two tasks: how the position in a network topology of each module can be defined using only its local connection sensors, and how these coupled modules collaborate their local actions to accomplish global effects such as locomotion and reconfiguration.

1.3. Survey

This section shows a recent survey about research in modular robotics. It focuses on real physical applications. Table 1.1 – 1.3 show the main technical specifications of the most important developments in this huge field. The most terms should be self-explanatory.

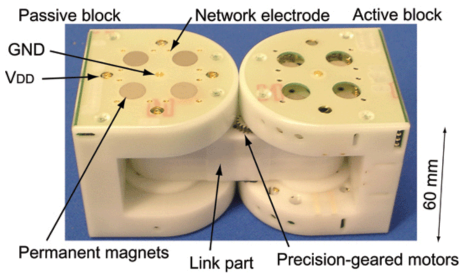
The robots that are listed here represent 3D structures. That means, they can move in all three spatial dimensions. More about 2D structures like the *Metamorphic Robo*t by Pamecha [19] and *Fractum* by Murata [16, 29] in the listed citations.

	homo-geneous	lattice/chain	self-reconfigurable	docking/dedocking	control-strategies
M-TRAN II	yes	lattice/chain	yes	magnetic force/SMA, springs	centralized, distributed (CPG+GA)
Polybot(G3)	yes	chain	yes	mechanical/SMA	centralized (gait tables)
CONRO	yes	chain	yes	mechanical/SMA	master-slave, decentralized ("hormones")
The Crystalline	yes	lattice	yes	mechanical (DC motor)	centralized, distributed ("PacMan")
The Molecule	yes	lattice	yes	mechanical (DC motor)	centralized
Telecube	yes	lattice	yes	magnetic force/SMA	–
I-Cubes	yes	lattice	yes	mechanical/SMA	centralized
Fracta	yes	lattice	yes	mechanical (DC motor)	distributed

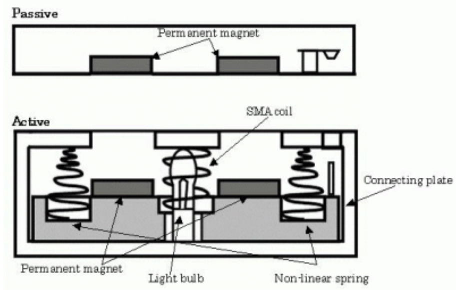
Table 1.1.: Global Properties of the listed modular robots

1.3.1. M-TRAN II: Self-Reconfigurable Modular Robot

M-TRAN II is the second prototype of the Modular Transformer developed by the Distributed Systems Design Research Group around Haruhisa Kurokawa. At time it is the most impressive modular robot concerning design, automatically connecting and disconnecting and locomotion. It is composed of two semi-cylindrical parts and a link working as actuator with two DOF. Even though the two actuated axis are aligned in parallel which only allows rotations in one plane, multiple connection surfaces compensate this disadvantage. The connection mechanism works with a combination of permanent magnets, nonlinear springs and SMA actuators. Magnetic connectors have advantages in accuracy during approach, but the polarity of magnets leads to male/female design. Figure 1.2 shows a M-TRAN II module and a figure of the connection mechanism [15, 12, 36].



(a)



(b)

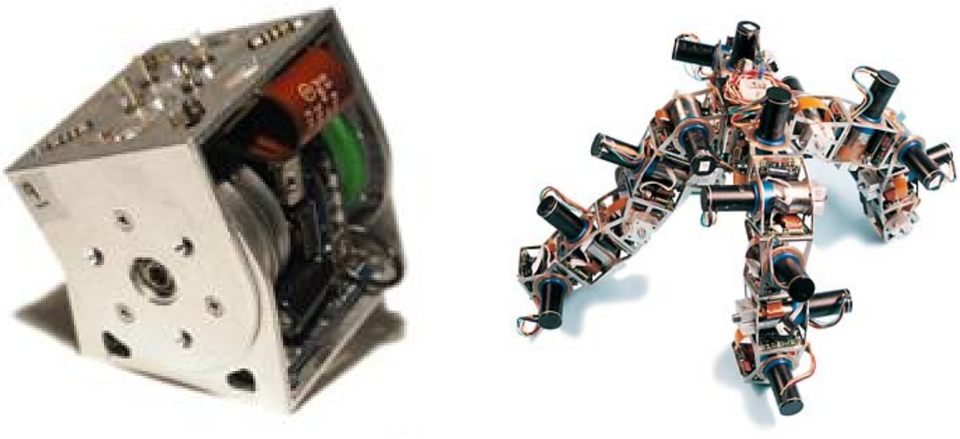
Figure 1.2.: M-TRAN module (a) and connection mechanism (b)

M-TRAN II is controlled centralized. One can set together any configuration on the PC. It is checked for connectivity and collision. A motion planner for lattice-like flow was developed working with the help of a motion-rule database making the cluster follow a desired trajectory. For ”real” locomotion a combination of a neural CPG[2] and GA[3] for optimization were proposed. A number of locomotion patterns were derived, which were also experimentally verified [35, 15].

[2]abbr.: Central Pattern Generator
[3]abbr.: Genetic Algorithm

1.3.2. Polybot: A Chain Reconfiguration Robot

Concerning mechatronics, integration, miniaturisation and use of MEMS technology, *Polybot* (mainly generation 3) can be seen as the best developed. As shown in table 1.2 it has high-tech motors, plenty of sensors and fully hermaphroditic connectors. Even the communication is established via CAN (Controller Area Network) bus a high-performance bus communication for distributed control. However, all this high-tech has one big disadvantage caused by these numerous devices. There are no batteries included therefore it can not work untethered. Figure 1.3 shows the latest prototype and legged configuration of the former generation. The connection mechanism works with a "latch in the hole" principal (90° increments in orientation possible) and SMA actuators for locking. But with two connection surfaces per module only snake-like configurations are possible (without help of passive elements, as seen in the humanoid-videos) [13, 34] .



(a) (b)

Figure 1.3.: Polybot G3 prototype (a) and G2 in spider configuration (b)

The control of *Polybot* is centralized, just like M-Tran. Their modules consult a gait-control table, to get unique messages tailored for each module. For this challenge they propose a description method (NML) for describing actual configurations and Phase Automata as a kind of event-driven state machines. Polybot seems to be the most

	DOF	"ability to move itself"	connectors	actuators	gears (max. torque)	size [mm]	weight [g]	water-proof	material
M-TRAN II	2 (rot)	no	male/female	DC-motor	reduction (190Ncm?)	66x132x66	400	no	Delrin®
Polybot(G3)	1 (rot)	no	hermaphroditic	Maxon-pancake (brushless)	planetary+Harmonic Drive (150 Ncm)	50X50X45	200	no	–
CONRO	2 rot= pitch+ yaw)	yes	male/female	Servos (Futaba S3102)	incl. (40 Ncm)	length=100	115	no	Delrin®
The Crystalline	2 (lin.)	no	male/female	toy-mini-motor (21 Ncm)	rack and pinion	178x51x51	341	no	–
The Molecule	4 (rot)	on partner with opposite sex	male/female	Servos	worm gear for connectors	102x140x102	1400	no	ABS plastic
Telecube	3 (lin)	no	male/female	Maxon-DC-gear motor	lead screw (12N)	60x60x60	300	no	Ertalyte®
I-Cubes	3 (rot)	no	male/female	Cirrus CS 21-BB Hi Perf. Servo	worm gear	length=80	370	no	rapid prototyping plastic (FDM)
Fracta	6(12) rot	no	hermaphroditic	DC motor	timing belts, worm gear, Harmonic Drive	length=265	7000	no	–

Table 1.2.: Mechanical Properties

	sensors	computing power	communication	power consumption	ext. power/ batteries	price
M-TRAN II	–	BasicStamp II	serial (4800 baud,async.)	4 W	yes/3.8 V, 700 mAh, Li-Ion	–
Polybot(G3)	HALL(joint angle), 4 accel., force, 4 IR	Motorola PowerPC 555+ 1 MB ext RAM	CANbus	–	yes/no	–
CONRO	IR, CMOS camera	Stamp II(e)	IR-serial (9600 baud)	1.9 W	yes/yes	–
The Crystalline	IR, HALL (exp./contract.)	Hitachi HD64F3644H)	serial, IR	–	yes/3V Li-Ion	300 $
The Molecule	–	microprocessor	no intermolecular, RS-485 with workstation	–	yes/no	1000 $
Telecube	elctr. contacts, IR	–	IR	1-2 W	yes/no	–
I-Cubes	mechanical encoders, contact switch	microprocessor (H11)	no intermolecular, RS-232 with PC	–	no/yes	–
Fracta	6 contact sensors, rotary encoders	–	–	7 W	yes/no	–

Table 1.3.: Electronics and Communication

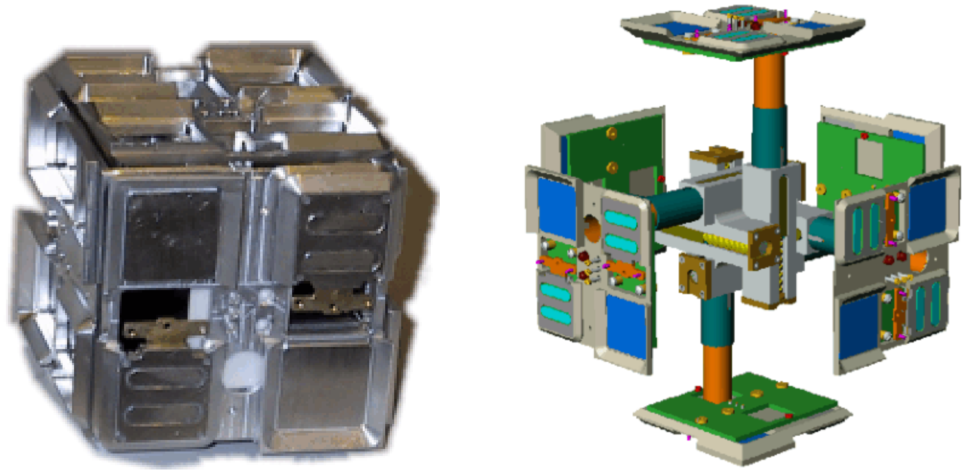
skilled modular robot in mechatronic aspects. But the problem of control in distributed systems is inadequately solved by Polybot [33, 38, 37].

1.3.3. Telecube

A linear actuated lattice-like robot called *Telecube* was developed by the group around Yim at PARC. Like Polybot it shows mechatronic design as its best. Custom made lead-screws (with telescoping tube linear actuators), different low-friction and light weighted plastics and a combination of magnetic- and SMA-actuated connectors affect this design (see Figure 1.4) [28, 31].

1.3.4. CONRO

CONRO was developed by a group around Wei-Min Shen and Peter Will at the University of Southern California in Los Angeles. Every module has two standard RC servomotors corresponding to pitch and yaw and it is self-sufficient and autonomous, because it has all computation hardware, batteries and sensors on-board. A single module

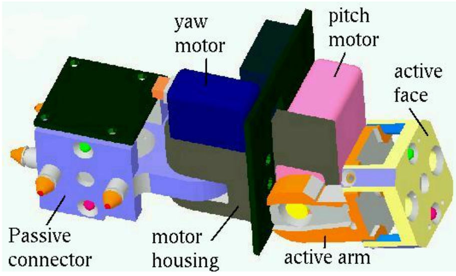


(a) (b)

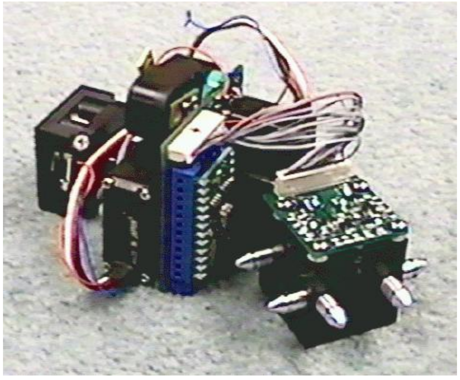
Figure 1.4.: Telecube fully contracted (a) and CAD drawing of the design (b)

is able to wiggle its body (limited ability to move). An equation for the ratio between module-size, weight, capacity and power consumption is proposed and considered during the design. This leads to a continuous operation time of 35 minutes and the power of the servo to lift one other module [4]. The docking works fully autonomously. It is divided into three steps: open loop phase for moving the IR-sensors into a line of sight, the closed loop phase with the help of the IR receiver signal and the inverse kinematics and at last the entrapment phase to push the pins in the sockets and latches, performing high-frequency small "shaking" movements to overcome friction [26]. It needs 3 minutes with a successful ratio of 80 % to connect from random snake structure. A disadvantage is that the connection mechanism between the elements is not genderless. For more information concerning the hardware refer to [3, 5].

For continual rediscovery of network topology an Adaptive Communication (AC) protocol is proposed. Every possible combination of connections can be crypted by a variable (all together 33 for a *CONRO* module). This information is provided by the IR sensors in the connection plates. When every module probes all its connectors in every cycle of the AC routine the topology is known in at most two cycles, independent whether it is a acyclic (root-tree-model) or a cyclic graph. For locomotion or reconfiguration the concept of hormones, which one finds in many biological organisms, comes in to play . A hormone is a special message which carries either direct servo-values or higher-level



(a)



(b)

Figure 1.5.: The CONRO element as CAD design (a) and real prototype (b)

sequences which can trigger the servos or latches to accomplish, for example, a detaching action. This variable passes every module, but it will be received only by those according to a rulebase. A rulebase can be seen as a gait control table (or a set of instructions for reconfiguration), but used for every configuration. For example, it does not matter if a snake consists of four or 40 modules or if a hexapod ”loses” legs becoming a quadruped. Additionally, the communication is much faster than a centralized control and it has the advantages of being fault-tolerant, asynchronous, scalable.The questions remains how to develop the appropriate rulebase for a particular global behavior automatically [24, 25, 27].

It should be remarked that according to the homepage of the lab[4] a new generation of self-reconfigurable modules named *SERE* is planned.

1.3.5. The Crystalline Robot and The Molecule Robot

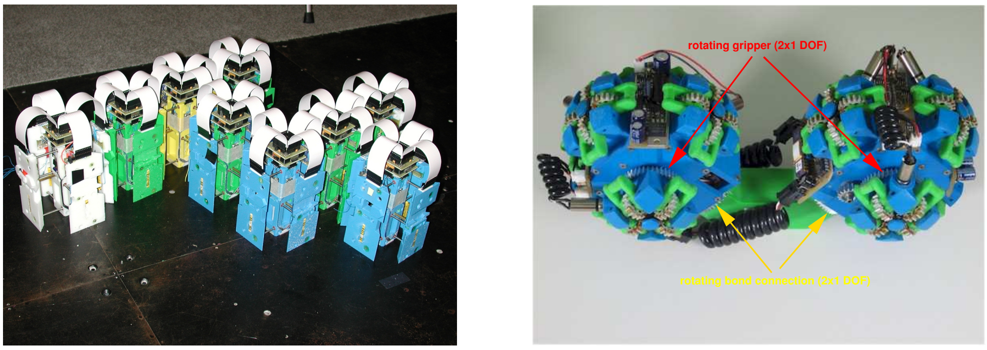
Both robots are the result of projects in self-reconfiguring robots conducted by the team of Daniela Rus at Dartmouth Robotics Lab.

The Crystalline Robot is a two-dimensional self-reconfigurable modular robot system composed of Atoms. The Atoms are square and actuate by expanding and contracting by a factor of two in each dimension. It is together with Telecube, the only modular

[4]http://www.isi.edu/robots/sere/

robot in this survey with linear actuation. This makes it suitable for walking through volume (not on the surfaces of a cube of modules, rather tunneling it). Respectively two of the four faces contain passive (channels) or active (bars) connectors that lock by a quarter-turn of the bar [2, 22, 23].

Describing the *Molecule Robot* is not uncomplicated, especially finding the actuated parts and getting an idea about possible movements. Like everywhere in our universe a molecule consists of atoms, in our case two, which are connected with a rectangular bond. Every atom is able to rotate around this bond 180 degrees. Hereby, the right-ankled bond leads to movements in 3D. Additionally, the only axis of the atom which is perpendicular to the bond-connecting face and lies not in the same plane like this, can rotate also 180 degrees. The connectors used for the Molecule Robot can be compared to the gripper mechanism of Swarm-Bot[5]. A non-backdrivable worm gear mechanism drives gripper arms (male) to connect to certain corresponding points (female) on the surface. *Molecules* either have all male components or all female components as connectors. This design allows elements to move lattice-like only on a substrate with a contrary sex then itself. Figure 1.6 shows the last prototype of *The Crystalline Robo*t and a "male" *Molecule*.



(a) (b)

Figure 1.6.: The Crystalline Robot (a) and The Molecule Robot (b)

To control *The Crystalline Robot* in a decentralized way the PacMan algorithm is proposed. They took the concept of pellets (food in the PacMan video game) which code the path each module has to follow in order to achieve global lattice-like motion. However,

[5]Swarm-Bots: European Project (incl. EPFL/Lausanne) to study new approaches to the design and implementation of self-organizing and self-assembling artifacts. See: http://www.swarm-bots.org/

the completeness of the algorithm relies on a convex shape of the current configuration which is difficult to maintain during locomotion [1].

Due to the L-shaped construction of *The Molecule Robot,* it is capable of convex and concave 90-degree transition, which in practice allows the module to walk on a surface and when it reaches a wall, to overcome the transition and cross the wall. A lot of planning algorithms for configuration and trajectory planning have been applied to this robot (incl. parallelization and scaffold algorithm). For more information refer to [11, 10].

Because of their lattice structure and behavior they can not yet show impressive movements like chain robots, but they seem to be a very good platform for studying motion planning, parallel computing and path optimization.

1.3.6. I-Cube

Ünsal and Khosla [30] have designed *I-Cube* a modular self-reconfigurable robotic system. *I-Cube* is a partite system composed of a three DOFs link and a passive element as connector (see Figure 1.7). In the right part of the figure you can see the attachment capabilities: (a) move from one cube face to another, (b) move one cube while attached to another and (c) move from one cube to another. The connection is similar to the Crystalline Robot, instead of a bar a cross-shaped male/female mechanism is used.

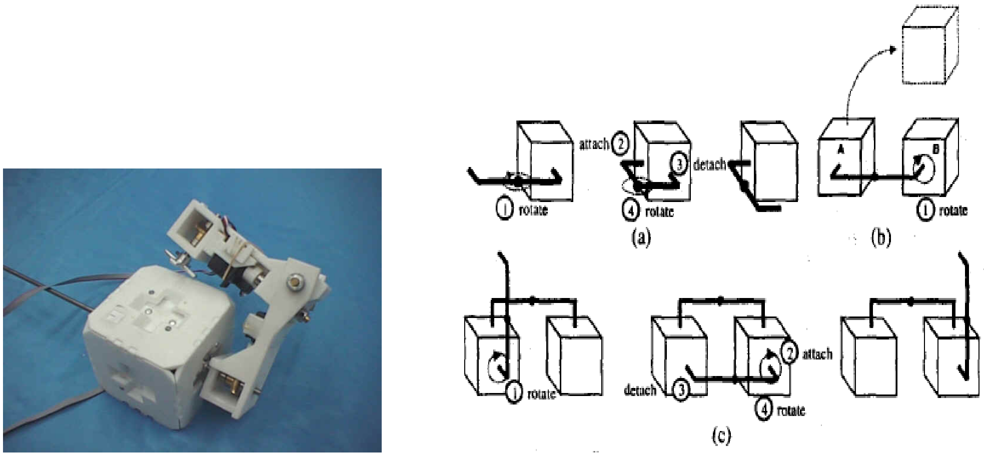


Figure 1.7.: I-Cubes module (left) and examples of link motion (right)

This construction tends to discover new possibilities in design and actuation. The approach for a moving link is unique in this field. But one can not proof the current status of this project and it is the only one where no videos exist, showing real robot motion. More informations about control algorithms (incl. metamodule and multi-layered planner) can be found in [20].

1.3.7. Fracta

Fracta is the 3D version on Fractum and made in 1998. It can be marked as the first feasible 3D system with self-reconfigurability. In this system a unit has 6 rotational DOF and 6 actuated connection mechanisms. These 12 DOFs in total are driven by one DC motor embedded in the central body. Twelve electro-magnetic clutches are used to connect the motor output axis to one of the DOF. A junglegym-like structure is built by this unit, and it is reconfigurable by appropriate sequence of connection/disconnection and rotation motion among the units (in or out of plane). The connection mechanism is a gripper-like but genderless construction.

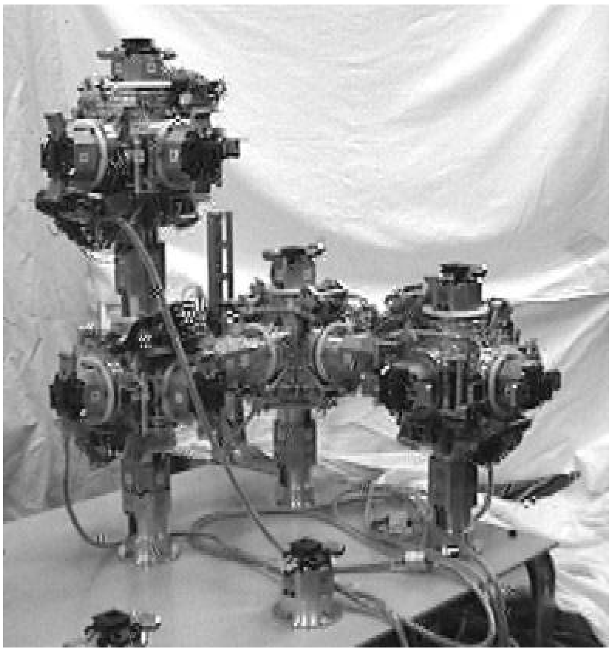


Figure 1.8.: Fracta modules (3D)

Because of the design (weight =7 kg and size) it was not built for impressive movements. The more interesting was to study distributed algorithms, including description methods of the current structure and path planning (stochastic relaxation and heuristic algorithms).

The drawback of this design with high DOFs is that each module is quite big and heavy.

1.3.8. Atron (Hydra-Project)

Atron was built by the Adaptronics Group, University of Southern Denmark, as part of the HYDRA project funded by European's Information Society Technologies (IST) programme[6]. Atron is a lattice-based system and it is built up by two half-spheres that are able to rotate relative to each other around an axis that goes through the centre of the module (see Figure 1.9). The elements have tangential connectors, four on every half-sphere. Around the male/female mechanisms the casing is flattened for better alignment during connecting. Changes in configuration are achieved by perpendicular alignment of rotation axis's of the elements performing 90 degree rotations around the equator. Communication is based on infra-red devices integrated in the connectors.

(a)

(b)

Figure 1.9.: Atron physical prototype (a) and CAD model of attachment mechanism (b)

From the controlling point of view there is only particular information. Some keywords are: rule-based controllers, metamodules, virtual force-, gradient- and genetic algorithms [6, 17, 18].

[6] www.hydra-robot.com

1.3.9. Titech

At the Tokyo Institute of Technology one can find a lot of research in modular robotics. Not really self-reconfigurable, but modular, the famous ACM-snake by Hirose [14] was developed at the same institute.

Another department created a *pneumatic* cellular robot, a lattice-like robot with bending bellows as actuators mounted on every surface of the cube-shaped body. The bellow is enclosed by a custom-made stabilization mechanism which allows 180 degrees rotation in the two-dimensional plane. Also, they developed a selective valve mechanism to reduce the number of mechanical elements. They showed automatic reconfiguration but the system is notable only because of the interesting actuator [8].

Another cellular robot with linear actuation and sliding mechanisms is proposed by Inou [9]. They implemented a two DOF slide motion mechanism, which one can imagine in the real world as cubes sliding over their edges and showing different structures, even under load.

1.3.10. Fractal Robots

The homepage of the Fractal Robots project[7] shows ”impressive”, blurred animations of their real robots. The elements are shiny aluminium blocks which seems to have no connector or mechanism - but they slide magically over another. By the way, one can buy this robot for 1000 $ per cube. For me it seems a little bit like a hoax, because no papers about this project can be found and I discovered several topics in news groups, having discussions about it. Figure 1.10 shows the cubes in different sizes.

[7] http://www.ecu.pwp.blueyonder.co.uk/

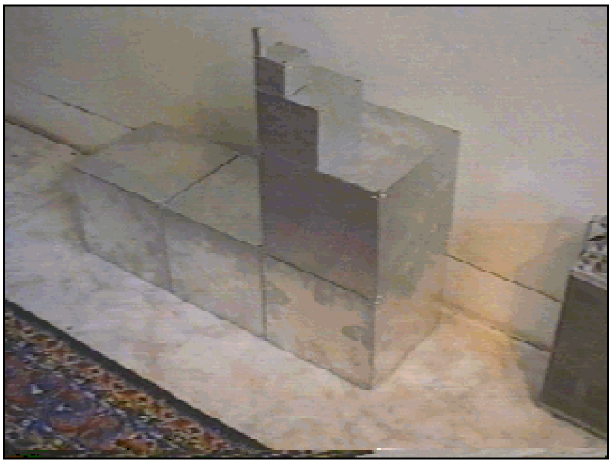


Figure 1.10.: Fractal Robots

2. Characterisation and Design

The general goal of this project, was to characterise, design and realise a modular robot unit. To determine functional specifications we made a list of properties similar to tables 1.1 – 1.3. Table 2.1 shows the specifications for the first prototype.

Specifications	**BIRG-ModRob**
homogeneous	yes
lattice/chain	chain
self-reconfigurable	no
DOF	1–2
"ability to move itself"	yes
computing power	FPGA
communication	Bluetooth
price per module	50–70 €

Table 2.1.: Specifications of BIRG-ModRob

Additional constraints are listed as follows:

- using of as much off-the-shelf parts as possible
- make the module itself modular, in the sense of being able to change the design by replacing or adding parts
- "open-design" for electronics and sensors which still have to be implemented
- time = 5 months for realisation of up to 10 prototypes

Especially the combination of the first and the last point in the list above made it a challenge. due to the problem of finding parts from any area that fit the design and could be delivered in time with a good price. The result of using off-the-shelf parts is a so called open-design, which means no determination on custom made pieces, rather changing the components easily.

2.1. Design Steps

2.1.1. First Draft

Because of the multiple constraints, I decided for a pretty simple construction with one degree of freedom (rotational) inspired by the shape of M-Tran II (see section 1.3.1) and the mechanism of WormBot [7]. The primary motivation which points towards a rotary drive is the chain-like behavior of the final structure. Added to this, that linear movement can only be achieved by gears (rack-pinion and worm gear, lead-screw) or by direct actuators (piezo, pneumatic-cylinder). Each of this possibilities can not satisfy the given constraints, especially price and time. In Figure 2.1 you can see the very first idea consisting of a main body with a round shaped side and an u-shaped profile for transmission of the torque.

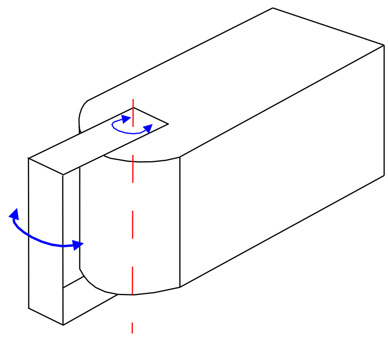
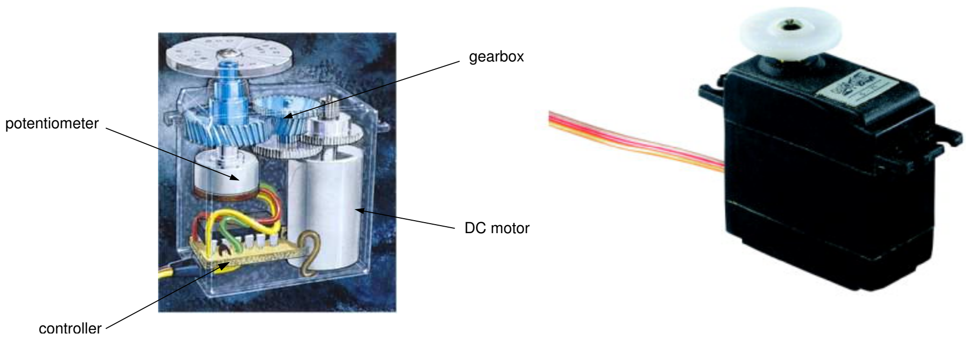


Figure 2.1.: First draft of the element consisting of body and u-profile

2.1.2. Actuator

Based on the result of the first design step we needed a rotary actuator with a good ratio between torque, price, weight and size. I decided for an electromagnetic actuator because working range, torque/weight-ratio and the availability in plenty of different shapes, sizes and designs. From the huge field of electromagnetic drives I furthermore decided to use standard RC-servos as known from airplane or car-models. I also considered DC-motors in combination with a gear box which are in advance in terms of size and shape when you are looking for some ”closed” integrated solution. Stepper motors have to be taken in account too, but I did not find some small device in the given price-range.

RC-servos are used quite often in robotics. They have the advantage of being optimized for different requirements concerning torque, size and weight. The most important feature is the integrated position control consisting of potentiometer, controller (Mitsubishi M51660L) and power circuits. Figure 2.2 shows the main parts. This leads to three inputs which are power supply (DC, 5–7 V), ground and a pulse-width modulated (PWM) controller signal.



(a) (b)

Figure 2.2.: Main parts of a standard RC-servo (a) and servo S-71 used in the module (b)

The PWM-signal has two important parameters: pulse length and frequency. The frequency is either 20 Hz or 400 Hz and the pulse-length determines the position directly via its zero-position value (mostly around 1.5 ms) and resolution (mostly 10 μs per 1 °). From the work with different servos, even from the same brand it is known, that these parameters are not the same and have to be calibrated for every servo. This does not really support the homogeneity and the exchangeability in modular robotics. To overcome this, parameters should be saved in any device directly according to each servo. Another disadvantage is the box-like shape which is pretty standardized (recently one can find so called "flat-servos"; see Figure 2.3) and therefore affect the whole design the most.

When you have to choose from the enormous amount of different servos and brands there are two parameters which appear to be most important: maximum torque and price. Due to the maximum torque is the weight because of metallic gears and bearings (for higher torques) and the size (bigger motors). The price is almost linear to those properties. To make a first calculation about the required torque you have to have first estimations about size and behavior in later configurations of one element in the modular robot. The latter means how far one element should be able to lift one other or more

Figure 2.3.: Flat RC-servo

of the modules. The size is a very important question which can not be fully answered yet in this early step of design. Therefore, a small calculation in has been done which takes in account the measurements, weight and options to lift up to three other elements (see Appendix). This result does not take into considerations moment of inertia and acceleration for dynamical behavior but which means not a big error due to the size of the modules and the speed. The results of the calculation show a range from 9.86 Ncm (lifting one module) to 73.8 Ncm (lifting three modules). Especially, the length which means the distance from the center of rotation (motor shaft) to the center of mass of the other element has the biggest influence on the required torque which should be considered in following optimizations of this prototype. Table 2.2 shows a selection of servos with needed torque range and price.

	brand / type	max. torque [Nm]	price [€]
1.	FMA Direct PS705MBB	62 Ncm	25.-
2.	Simprop SES 640 2BB MG	64 Ncm	22.-
3.	Servo S-71 J/R	73 Ncm	19.95
4.	Kopropo PS 2174 FET	120 Ncm	70.-

Table 2.2.: Selection of RC servos arranged by torque

After my investigations including the European, American and Asian market I chose no. 3 from the table. Because of its torque, full metal gears and metal bearings (see Figure 2.2 (b)). A current special offer at Conrad Electronics, Germany for this type made it really number one. The following list shows the main specifications:

- max. torque: **73 Ncm**

- max. speed: **60°/0.16 s**
- working range: **180 °**
- size: **41 x 20 x 41 mm**
- weight: **58 g**
- price: **19.95 €**

I also tried to get information from different newsgroups, robotic boards and forums in the Internet concerning reports on quality and behavior in real world of this servo. But I did not find anything about the type I chose. In Section 2.2 you find more about experiments with the servo under different conditions, including a long run test.

At the end of my investigations I found a servo with quite impressive properties (no. 2 in Table 2.2). It has a similar torque like the one above, but it is smaller in size and weight. We tested two of them and they seem to be quite good, despite of the inaccuracy under upper range of load. At this point it makes no sense to have a much smaller solution, because of open questions like batteries and electronic components which still have to be integrated yet. But as an alternative for a second prototype this motor should be further analysed.

2.1.3. Casing

As the most important device for the module has been made, the fixation of the motor inside the module should be done. Because of the existing holes on the servo case made for screws to fix it on substrate, a sandwich-design is proposed to achieve shape and volume. As the main material for the case I chose printed circuit board PCB-material (thickness = 2.4 mm), due to its material properties like stiffness, weight and available manufacture technologies in workshops here at the EPFL. You can directly export your data from your computer aided design (CAD)-tool to process them rapid-prototyping-like into physical matter. I designed two plates, one with a cut for the servo and holes for the fixation, hold in place by two hexagonal distance bolts made from Polyamid. The shape of the plates and the height of the bolts determine surface and volume of the element (see Figure 2.4). In this step 3D-CAD-tools, like ProEngineer ® show their advantage in construction and design because one can change dimensions virtually but getting a clear impression about the influence of the overall design. We intended to have scaled dimensions which means to have width, height and depth as a multiple of a what we nearly realized (a = 45 mm and b = 2a, c = a).

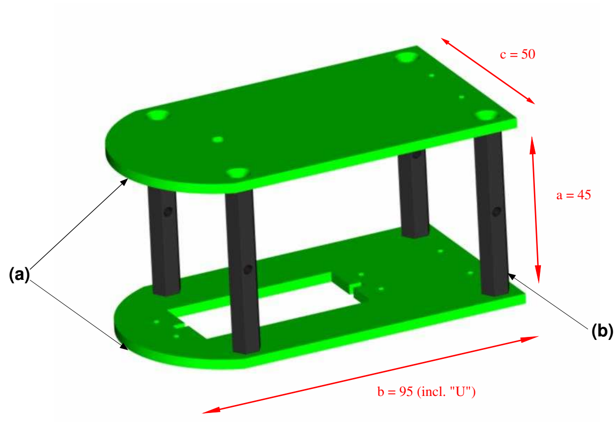


Figure 2.4.: Sandwich design of the casing: distance bolts (b) and milled PCB plates (a)

At last the box has to be closed to have possibilities for intermodular connections and for protection. This is achieved by simply adding plates from the same material (thickness = 1.6 mm) sidely directly screwed into the plastic distance bolts (with some custom metal thread made by HELICOIL®) and at the other with the support of small aluminum profiles (see Figure 2.5).

To fix the aluminium u-profile properly, one side is screwed to normal plastic servo-lever directly on the motor shaft, whereas on the other side an aluminium lever is fixed, rotating the construction around an axle . For better sliding friction a bronze bushing is pressed into the alu-lever. To keep the axle aligned with the servo shaft it is fixed in a PCB plate on the head of the servo with the original screws of the servo-case.

2.1.4. Connection Mechanism

2.1.4.1. General Goals

The question of interconnecting robot elements can be seen as the key-question in modular robotics. A connection mechanism should therefore have the following properties:

- power efficiency (no power consumption when connected)

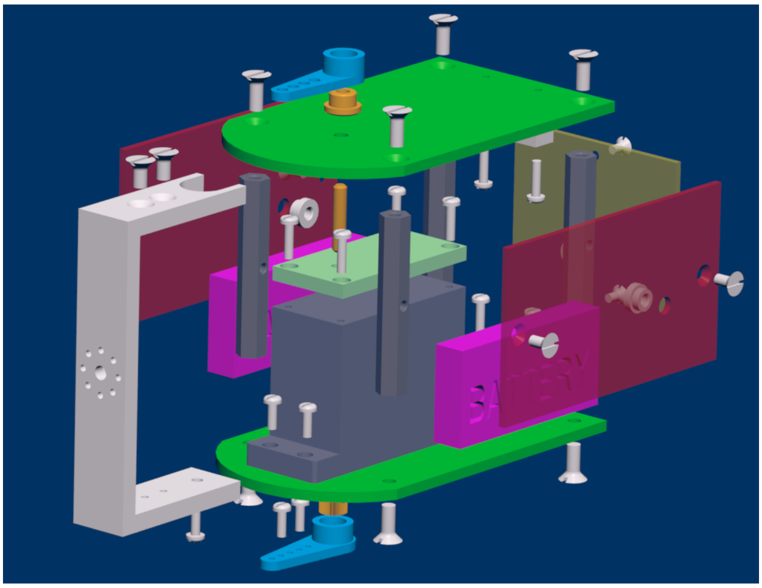


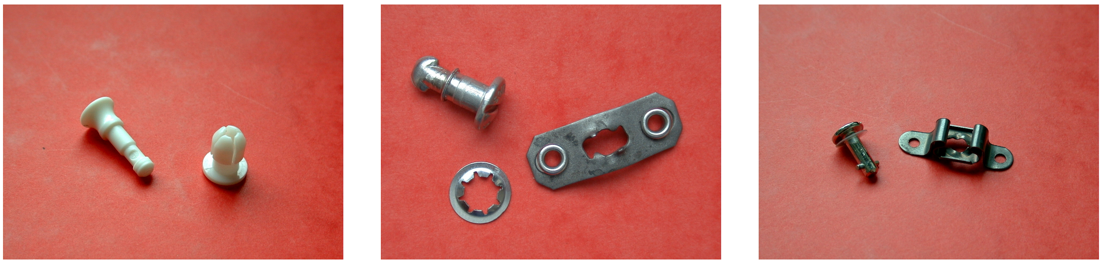
Figure 2.5.: Final Design of BirgModRob including batteries and motor

- reliability (connections must endure various operations)
- compact (mechanism must fit into a tight space)
- genderless (allows to connect every connector among one other)

Most of the existing connection methods are based on penetration and shape matching, where a latch is forced to slide into a groove or cut and is kept engaged by spring force. For disconnecting one needs some active force generated mostly by shape memory alloy (SMA) actuators (the only process which consumes power). The other option, you can find it in the M-TRAN system, relies on connecting via magnetic force and disconnecting with a help of a spring-SMA-actuator. The advantages of the mechanical solution are highly reliable links which can withstand lateral forces mechanically, in return they need very good alignment during the connection phase.

2.1.4.2. Realised Design

Additional to the general goals we wanted to have the possibility to interconnect the elements in any particular angle mainly with a resolution of 45 degrees and on several points on the surface of the module. Especially the last point is important for discovering structures useful for locomotion. The connector itself does not have to transmit any data or energy because of wireless communication and autonomous battery power. On the other hand, it is really difficult to find some off-the-shelf product as a connector which fits into a tight place and is strong enough to withstand forces and torques in all directions in space. We tried several fast locking mechanisms which you can see in Figure 2.6. For connecting you have to insert the male part into the cut and do a quarter-turn. The Nylatch® works by pushing the pin into the hole which is bending up the nylon-latches. But they do not fulfill the fact of being genderless. Imagine you mount the male part of those connectors to the alu "U" and the cam-locks inside the case - how do you attache two back sides of the modules together? The solution would be mounting male and female parts on every surface. However, you get into conflict with the goal having different orientations.



(a) (b) (c)

Figure 2.6.: Different off-the-shelf mechanisms: Nylatch (a), Southco® (b) and Camloc (c)

Finally, I decided for a combination of Velcro® fastener, a very strong tissue which works with principal of cocklebURS and screws. The Velcro® can stand stand up against sheer forces up to 600 N. Hence, you can fix up to three modules only with the Velcro®, especially when the surface is big enough like if you connect the sides of the modules. Screws which support any complicated structure can easily be screwed into some metal inlet with internal thread, which are pressed into the PCB plates. For first experiments you can now achieve a reliable connection with an average mounting time of 45 sec per connection.

Surely this is far away from having automatically connectible joints, but when first experiments with this design show sufficient results and when all electronics are included, it should be possible to create some custom made mechanism. Figure 4.1 shows two modules, attached with the current connection mechanism.

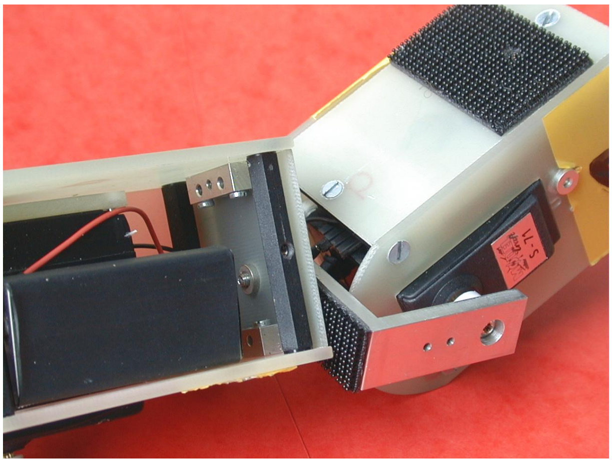


Figure 2.7.: Two modules connected in 45 ° orientation

2.1.5. Batteries and Electronics

With the question of batteries, which is an important issue for all autonomous robots, I profited by the work which has been done for the salamander robot at BIRG. They use rechargeable Li-Ion batteries with a nominal voltage of 3.7 V and capacity of 700 mAh. With this capacity the module could theoretically work up to 45 min depending on the load and the power consumption of the electronics. For the module we use two batteries connected in serial to achieve an output voltage ranging from 6–8 V, which is slightly higher then the input voltage of the servo, causing high velocity of the motor. They are fixed laterally to the servo, aligned to the back of the element to shift the center of mass, having less distance to the next motor shaft (when connected in serial).

For the first prototype I implemented a row of plugs fixed to the servo case. One wire is used to disconnect the batteries from the servo, preventing loss of charge and to access every battery individually for service and maintenance. The other plugs are used for monitoring the battery voltage and to connect the off-board controller. Figure 2.8 shows the open module with the batteries and a wiring schematic.

For the actual prototype the communication works as follows: any trajectory is created on the PC (refer to next chapter) and transmitted via the serial interface to the Bluetooth node. This device communicates with another BT-node which is connected to the FPGA, which further on creates the PWM signal for the several servos. From the FPGA you need two wires (signal and ground) leading to each module. In Figure 2.9 you can see the schematic of this open loop control.

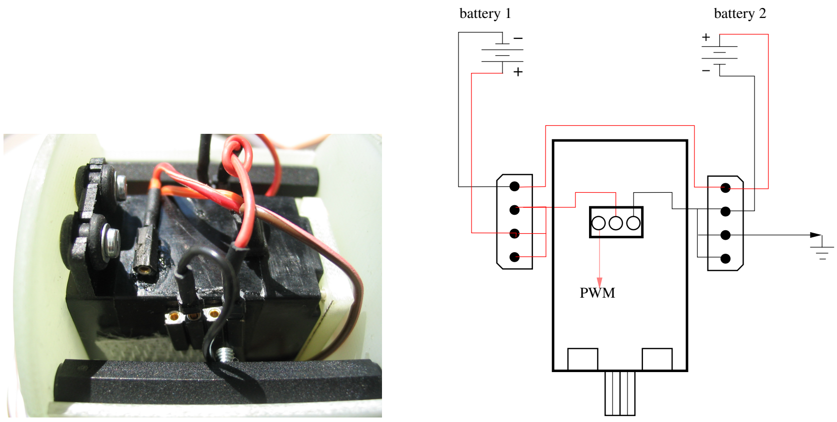


Figure 2.8.: Electronic connectors and wiring schematic

At the moment another student in the lab is working on the design and implementation of all electronics (power-management, Bluetooth and FPGA). When this is finished the modules should be completely autonomous (in the sense of power and communication) and intermodulare distributed control via Bluetooth should be feasible.

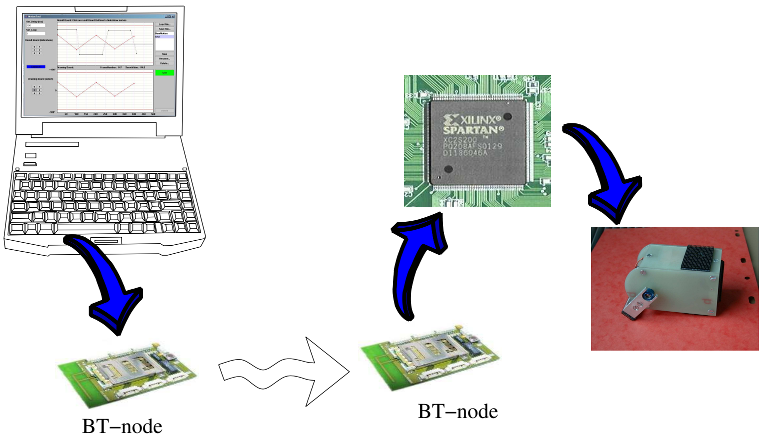


Figure 2.9.: Communication with off-board electronics

3. Software

3.1. Actual Control

Because of the absence of sensors we have a straight forward open loop control. The trajectories are created heuristically by trial and error (experiments show good results even with very simple input). We implemented two possibilities to control the robot.

3.1.1. Control File

We propose a trajectory file with a self explaining syntax which can be seen as a gait control table used in Polybot 1.3.2. Trajectory means a sequence of angle values over time for each servo separately. We do not process in absolute time values, instead we use fixed time steps ($\triangle t$). This prevents the file from becoming too complex and we think for our main purpose to study locomotion it should be more important to have looped movements instead of a highly resolved trajectory. The following shows the syntax of this file, named *traj.txt*:

*/*Description of the movement or configuration */*

<receiver>␣*(addresses of the modules separated by space)* ␣**</receiver>**

<set_delay>␣*(time step in ms)*

<set_time_factor>␣*(factor multiplied with <set_delay>)*

<loop>␣*(address of the loop)* ␣*(loop counter)* /*loop starts here*/

(motor values for each servo defined by <receiver> in the same order, separated by space)

</loop>␣*(address of the loop)* ␣*(value multiplied with time_factor every loop cycle)* /*loop ends here*/

We have given addresses to the loops (the area between <loop> and </loop>) to make it possible to interleave them. The *time_factor* sets any value, which is multiplied with the time step, to double or to halve the velocity of the movement. In Section 4.2 you can see several examples for such a file.

This file can be edited by hand (as was done during all the experiments) or export the angle values by any other program like Matlab®, Xfig or MS-Excel.

3.1.2. JAVA-GUI

For a more ergonomic way creating trajectories we wanted a graphical user interface where the user is able to simply create any desired trajectory by clicking on the screen. By chance a project at the lab had just been completed with the result of a GUI written in Java. Originally this tool was thought as a plugin for Webots[1] a platform for the International Robot Judo Contest[2] written by Jean-Philippe Egger. With the Java Motion Tool it is possible to create trajectories and to save them. The program has been modified to write the interpolated values into the above mentioned control file and to send it to the robot achieving a corresponding movement. Additionally you can connect the BT-nodes by pressing the "Connect!" button. Figure 3.1 shows a screenshot of this GUI. The board below is the drawing board and the upper board shows the already drawn trajectories.

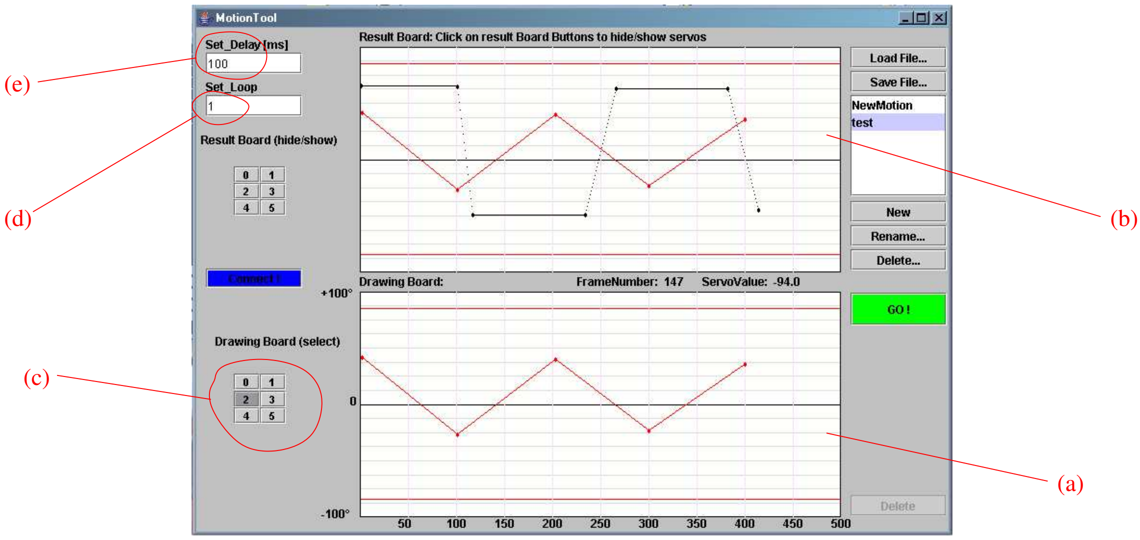


Figure 3.1.: Java Motion Interface for Modular Robot: (a) drawing board, (b) result board, (c) buttons to select the modules, (d) loop counter (see previous section) and (e) time step (prev. section)

The number of modules in the select area can be easily increased.

[1] a commercial 3D robot simulator by Cyberbotics Ltd. http://www.cyberbotics.com
[2] see: http://www.roboka.org

4. Experiments

4.1. Testing the Servo

To verify the specifications of the motor, test the accuracy and long run behavior, a test bench for the servo was constructed. Figure 4.1 shows the setup of the experiment.The main goal was to get information about the accuracy under load. This was done with the help of some printed semi-angular circle. The actual value was read of the scale. Three different experiments were done to have sufficient information about the motor in the future application. All experiments have been done under the same temperature and light. The servo voltage was fixed to 7 V.

4.1.1. Static Test

Several angles were every time approached from the 0° position. Unfortunately, there was no time driving every angle repeatedly to make statements about the statistic error. We measured the actual angle and compared it to the desired angle under different load. Additionally, the current was measured to keep the module in position. Figure 4.2 shows the results.

Figure 4.2 shows a satisfying accuracy up to a load of three modules. But even a maximum error of 7 degrees is not so much.

4.1.2. Dynamic Test

Here the servo was driving a sinus trajectory with adjustable amplitude and frequency. The actual amplitude was read off the angular scale at the reversal points. This experiment was done with a load of up to three modules, although with three modules only small amplitudes with low frequency could be approached. Figure 4.3 – 4.4 show the results. The non-coloured parts refer to canceled experiments to prevent the motor from destruction. These areas can be seen as no-go area for the parameters.

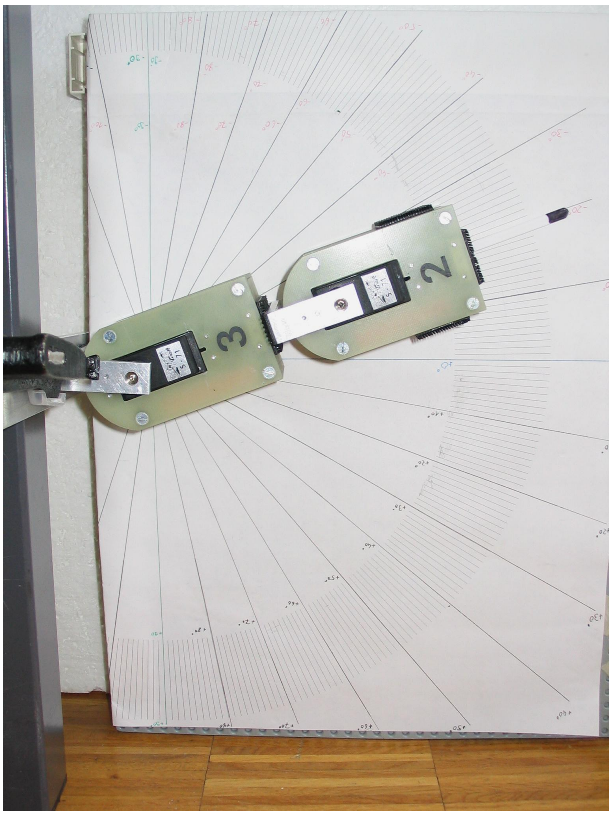


Figure 4.1.: Setup shows two modules and the printed angular disc

4.1.3. Long Run Test

A module was running for 7 hours without break, a load of two other modules doing a sinus trajectory with an amplitude of 50° and a frequency of 0.6 Hz. We discovered no abnormalities in movement and rhythm. A visual check up after the test showed no changes. Also a short static test was carried out which showed no significant bigger errors than before.

4.2. Tests with different structures with up to 6 modules

With the recent first prototype we were able to make the first tests of locomotion. Inspired by existing configurations, famous in modular robotics, we made snake- and inchworm-locomotion, several crawlers, polypeds and the rolling track. Surprisingly, we had pretty impressive results, although we had to cope with the problem of asymmetric friction - which is a key question especially in crawling locomotion. Fortunately, the shape (having multiple types of friction and coefficients) and some "intelligent" speed control (velocity dependant friction) could overcome this problem, so that we made the most experiments on standard hard wood floor. Following we will show some configurations and the control file. The figures showing screen-shots of videos we made and the corresponding trajectory file.

4.2.1. Modules moving by itself

This experiment showed, that the modules are able to move by itself. This is a big advantage and for a module with only one DOF, I have not found it during my research. For this experiment a Styrofoam base was necessary to achieve asymmetric friction.

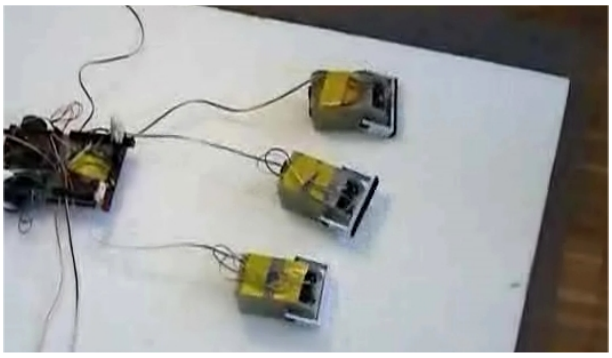
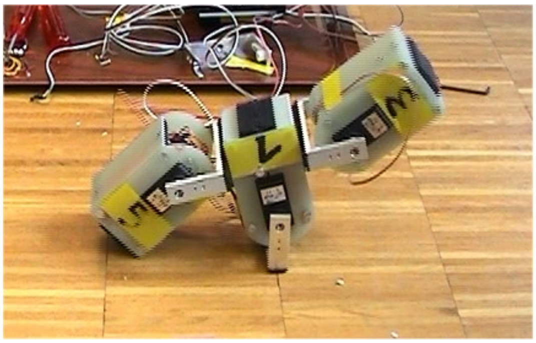


Figure 4.5.: Modules moving by itself pulling the controller board

4.2.2. Three modules

Picture 4.6 shows an interesting configuration with a passive pusher. This princip of locomotion was the most successful configuration, even in the other experiments. This indirect manipulation of inertia can help a lot overcoming the friction-problem.

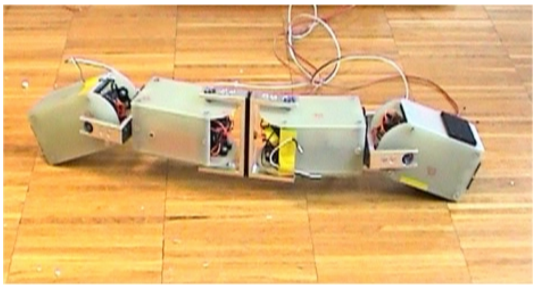


/*3-Heads2Side oft he middle*/
<receiver> 0 4 2 **</receiver>**
<set_delay> 170
<set_time_factor> 1.0
<loop> 1 30
0 0 0
20 45 45
</loop> 1 1

Figure 4.6.: Three modules hopping around because of the push of module no. 3 and trajectory file

4.2.3. Four Modules

As an example for a possible configuration you can see a sidewinder-like structure. This name was given because it moves laterally.

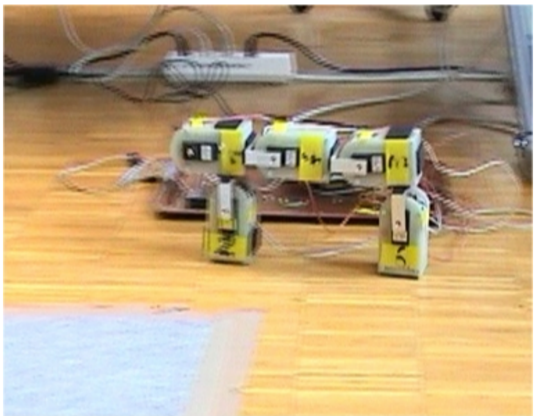


/* 4-Head2Head and Heads2Tail */
<receiver> 3 2 4 0 **</receiver>**
<set_delay> 100
<set_time_factor> 1.0
<loop> 1 35
-10 12 -12 10
-30 12 -12 30
-30 0 0 30
</loop> 1 1

Figure 4.7.: Four modules moving like a sidewinder and trajectory file

4.2.4. Five Modules

With five modules I tried a biomimetic way. Imagine a dog with immobile forelegs, jumping because of flection in the spine and hind limbs. This configuration brought very effective but pretty unstable locomotion.

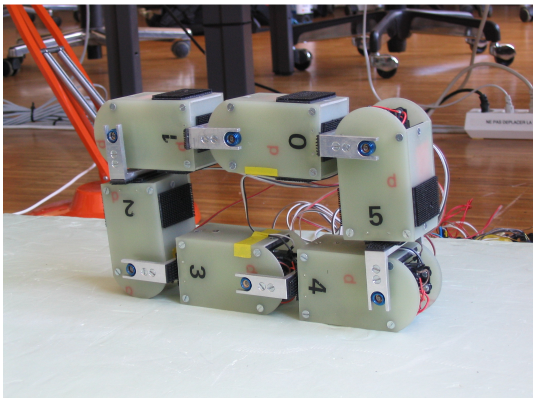


/* 5-Dog */
<receiver> 1 5 2 3 **</receiver>**
<set_delay> 200
<set_time_factor> 1.0
<loop> 1 10
0 -45 -45 0
0 0 0 0
</loop> 1 1

Figure 4.8.: RoboDog jumping through the laboratory and trajectory file

4.2.5. Six Modules

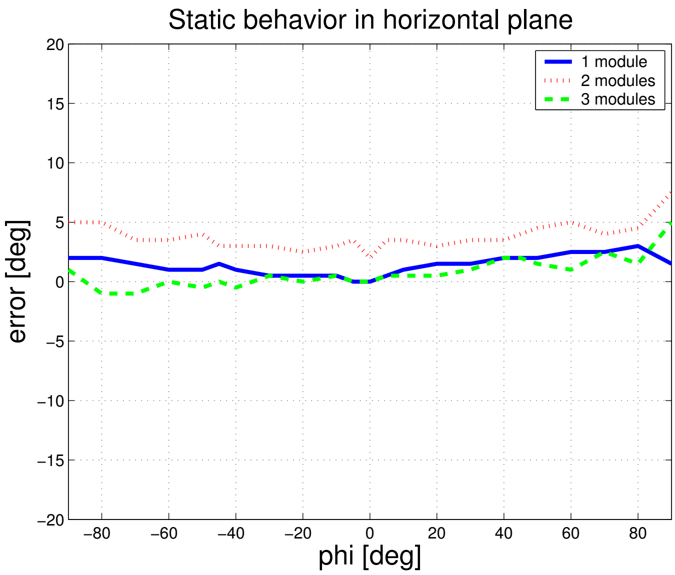
At last of the experiments we tried to make the rolling track, which has become pretty famous in modular robotics. The basic principal is, to shift a pair of 90°-values around the whole structure. Unfortunately we have only six modules, therefore the movement is not very smooth and sometimes the robot changed the direction. It has be mentioned, that especially with more modules (more DOFs), the heuristic method for creating the trajectories becomes very hard.



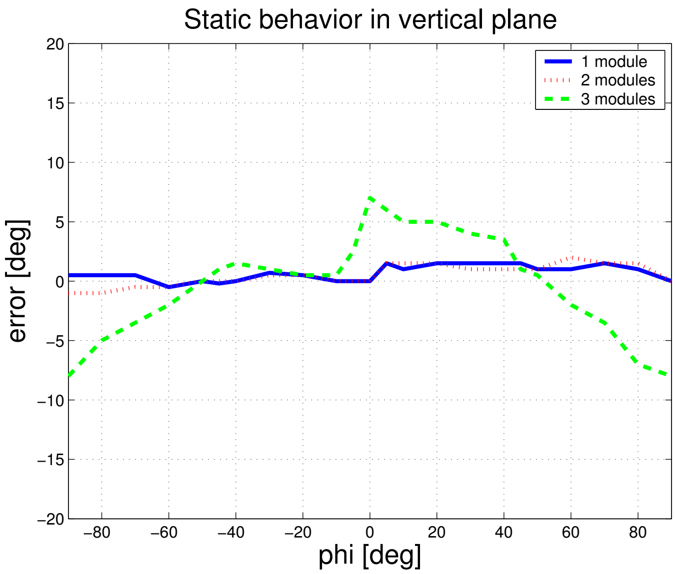
/* 6-Rolling Track */
<receiver> 0 1 2 3 4 5 **</receiver>**
<set_delay> 600
<set_time_factor> 1.0
<loop> 1 5
0 90 90 0 90 90
45 90 30 60 90 45
90 90 0 90 90 0
90 30 60 90 45 45
90 0 90 90 0 90
30 60 90 45 45 90
</loop> 1 1

Figure 4.9.: "Rolling Track" with six modules and trajectory file

All the here described experiments and much more interesting locomotion can be downloaded as video-files from birg.epfl.ch/page50163.html.

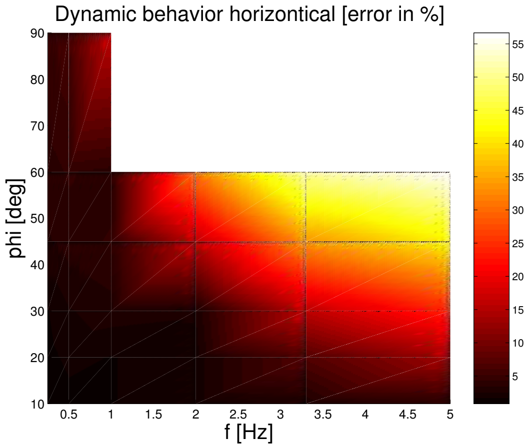


(a)

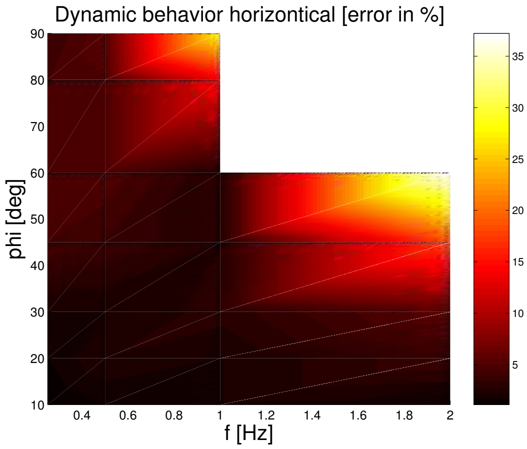


(b)

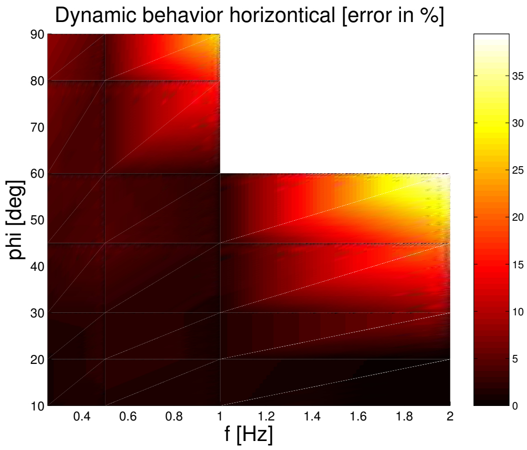
Figure 4.2.: Static Motor accuracy in horizontal plane (a) and vertical plane (b). Phi equals target angle. Figure (b) shows bigger errors due to gravity.



(a)

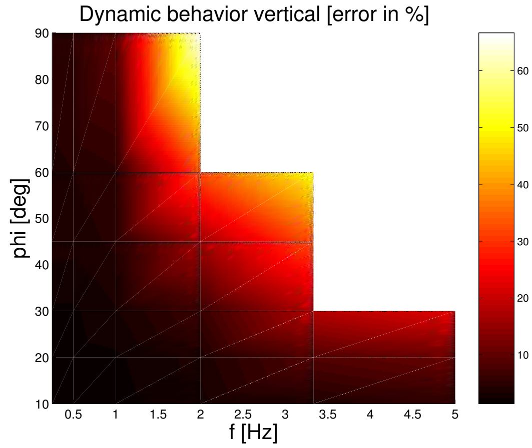


(b)

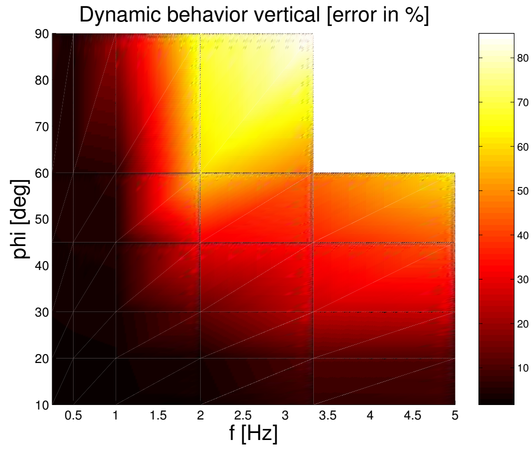


(c)

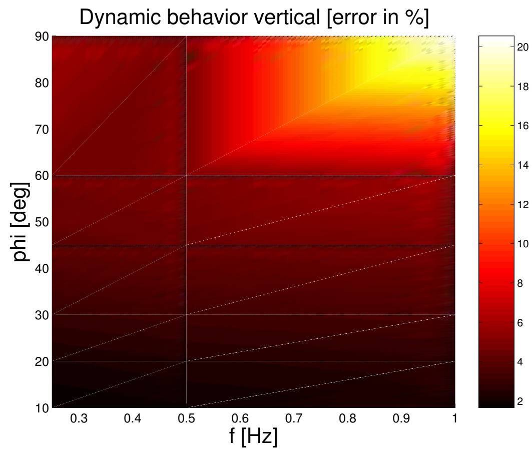
Figure 4.3.: Dynamic motor accuracy in horizontal plane. Load = 1 module (a), 2 modules (b) and 3 modules (c). Unfilled area shows canceled experiments due to exploding errors.



(a)



(b)



(c)

Figure 4.4.: Dynamic motor accuracy in vertical plane. Load = 1 module (a), 2 modules (b) and 3 modules (c). Unfilled area shows canceled experiments due to exploding errors.

5. Conclusion and Outlook

This report describes the different design steps and considerations that lead to the current design. Although some questions could not be fully answered the main goals of this project were accomplished. An ”open” design is proposed which is roughly fixed in form and function. The first experiments showed amazing results concerning the suitability for modular robotics. As long as the key question of automatic connector mechanisms remains, the robot can not be self-reconfigurable. Depending on the following applications the implementation of sensors should be considered. Most of these extensions should be feasible with more or less expense on the current design.

5.1. Proposals for the prototype 2

5.1.1. Actuator

We are mainly satisfied with the used servomotor. The first experiments showed that torque, speed and accuracy are sufficient. Especially the price makes it the best offer we could get at this time. But because the market of standard RC-servos is changing rapidly the offers should be checked frequently. In the appendix I put a list with the different brands I found. Going along with optimizations in size and weight the Simprop-servo (see: table 2.2) should be taken in consideration.

5.1.2. Casing

Depending on the integration of the Bluetooth, FPGA and the power circuit the optimization in size (especially the length) of the element should be taken in account. Decreasing the length would bring a much less torque for lifting a module. To win some space for sensors or additional circuits, the axle for the actuated ”u” should be directly integrated into the upper layer of the casing. Another possibility would be to make the ”U” smaller in length and to integrate it into the casing to have real plain surfaces over the whole construction. However, it has to be mentioned that most of the locomotion we showed is due to the ”active” support of this part.

5.1.3. Sensors

During the design process I was thinking a lot about information which one needs for experiments with locomotion and distributed control. Most of the sensors you find in modular robotics are due to the problem of self-reconfiguration (approaching the connectors) and communication. We are not in charge of these problems yet and so I was looking for other important data of the system which should be known.

Contributing to another project at the lab, which is about self-organisation in locomotion with the help of numerical search methods, it would be great to make real simulations with hardware in the loop. Therefore, you need information about the position of the robot over time. Absolute position could only be achieved by tracking sensors and orientation with gyroscopes or inclinometers.

Touch- or whisker-like sensors could be used to get information about collision between elements or stance status in a possible configuration. I discovered a piezo-sensitive foil[1] which could be attached to the corners or to the whole surface. Another possibility would be using small electret microphones with the help of intelligent signal processing.

One sensor already integrated in the servo is the position potentiometer of the motor. Attaching an additional wire can give either information about the actual position, or take its derivative for knowing if a desired position is reached. An additional feature like you find in Tobobo could be to manipulate the elements manually, to record the values and to replay the trajectory.

Which of these possibilities should be implemented remains a future challenge for the next prototype.

[1] Images SI Inc.

Bibliography

[1] Z. Butler, S. Byrnes, and D. Rus. Distributed motion planning for modular robots with unit-compressible modules. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 790–796, 2001.

[2] Z. Butler, R. Fitch, D. Rus, and Y. Wang. Distributed goal recognition algorithms for modular robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 110–116, 2002.

[3] A. Castano, A. Behar, and P. Will. The conro modules for reconfigurable robots. *IEEE Transactions on Mechatronics*, 20:100–106, 2002.

[4] A. Castano, W.M. Shen, and P. Will. Conro: Towards deployable robots with inter-robot metamorphic capabilities. *Journal of Auonomous Robots*, 24:879–888, 2000.

[5] A. Castano and P. Will. Mechanical design of a module for autonomous reconfigurable robots. In *Proceedings of IEEE/RSJ Intl. Conf. of Intel. Robots Systems*, pages 2203–2209, 2000.

[6] D.J. Christensen, E.H. Østergaard, and H.H. Lund. Metamodule control for the atron self-reconfigurable robotic system. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 685–692, 2004.

[7] J. Conradt and P. Varshavskaya. Distributed central pattern generator control for a serpentine robot, icann 2003. In *ICANN 2003*, 2003.

[8] N. Inou, H. Kobayashi, and M. Koseki. Development of pneumatic cellular robots forming a mechanical structure. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, pages 63–68, 2002.

[9] N. Inou, K. Minami, and M. Koseki. Group robots forming a mechanical structure-development of slide motion mechanism and estimation of energy consumption of the structural formation. In *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 2, pages 874–879, 2003.

[10] K. Kotay and D. Rus. Algorithms for self-reconfiguring molecule motion planning. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 2184–2193, 2000.

[11] K. Kotay, D. Rus, M. Vona, and G. McGray. Self-reconfiguring robotic molecule. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 424–431, 1998.

[12] H. Kurokawa, A. Kamimura, S. Murata, E. Yoshida, K. Tomita, and S. Kokaji. M-tran ii: Metamorphosis from a four-legged walker to a caterpillar. In *Proceedings of the Conference on Intelligent Robots and Systems(IROS 2003)*, volume 3, pages 2454–2459, October 2003.

[13] Yim Mark, Duff David G., and Roufas D. Polybot: a modular reconfigurable robot. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 1, pages 514–520, 200.

[14] M. Mori and S. Hirose. Development of active cord mechanism acm-r3 with agile 3d mobility. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1552–1557, 2001.

[15] S. Murata, A. Kurokawa, H.and Kamimura, E. Yoshida, K. Tomita, and S. Kokaji. M-tran: Self-reconfigurable modular robotic system. *IEEE Transactions on Mechatronics*, 7(4):431–441, December 2002.

[16] S. Murata, H. Kurokawa, and S. Kokaji. Self-assembling machine. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, volume 1, pages 441–448, 1994.

[17] E.H. Ostergaard and H.H. Lund. Evolving control for modular robotic units. In *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 2, pages 886–892, July 2003.

[18] H.H. Ostergaard, E.H.and Lund. Distributed cluster walk for the atron self-reconfigurable robot. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 291–298, 2004.

[19] A. Pamecha, C.J. Chiang, D. Stein, and G. Chirikjian. Design and implementation of metamorhic robots. In *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pages 1–10, 1996.

[20] K.C. Prevas, C. Unsal, M.O. Efe, and P.K. Khosla. A hierarchical motion planning strategy for a uniform self-reconfigurable modular robotic system. In *Proceedings of the IEEE International Conference on Robotics and Automation(ICRA '02*, volume 1, pages 787–792, May 2002.

[21] H.S. Raffle, A.J. Parkes, and H. Ishii. Topobo: A constructive assembly system with kinetic memory. In *Conference on Human Factors in Computing Systems, Vienna*, April 2004.

[22] D. Rus and M. Vona. Physical implementation of the self-reconfiguring crystalline robot. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2, pages 1726–1733, 2000.

[23] D. Rus and M. Vona. Crystalline robots: Self-reconfiguration with compressible unit modules. *Autonomous Robots*, 10(1):107–124, 2001.

[24] B. Salemi, W.M. Shen, and P. Will. Hormone controlled metamorphic robots. In *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, pages 4194–4199, 2001.

[25] W.M. Shen, B. Salemi, and P. Will. Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, 18(5):1–12, October 2002.

[26] W.M. Shen and P. Will. Docking in self-reconfigurable robots. In *Proceedings of the 2001 IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1049–1054, 2001.

[27] K. Støy, W.M. Shen, and P. Will. A simple approach to the control of locomotion in self-reconfigurable robots. *Robotics and Autonomous Systems*, Volume 44(3-4):191–199, September 2003.

[28] J.W. Suh, S.B. Homans, and M. Yim. Telecubes: Mechanical design of a module for self-reconfigurable robotics. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation Washington, DC*, pages 4095–4101, May 2002.

[29] K. Tomita, S. Murata, H. Kurokawa, E. Yoshida, and S. Kokaji. Self-assembly and self-repair method for a distributed mechanical system. *IEEE Transactions on Robotics and Automation*, 15(6):1035–1045, 1999.

[30] C. Unsal and P.K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. In *Proceedings of the IEEE International Conference on Robotics and Automation(ICRA '00)*, volume 2, pages 1742–1747, April 2000.

[31] S. Vassilvitskii, M. Yim, and J. Suh. A complete, local and parallel reconfiguration algorithm for cube style modular robots. In *Proceedings of IEEE International Conference on Robotics and Automation(ICRA '02)*, volume 1, pages 117–122, May 2002.

[32] J. von Neumann. *Therory of Self-reproducing Automata*. Univ. of Illinois Press, 1966.

[33] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans. Modular reconfigurable robots in space applications. *Autonomous Robots*, 14(2–3):225–237, 2003.

[34] M. Yim, Y. Zhang, K. Roufas, D. Duff, and C. Eldershaw. Connecting and disconnecting for chain self-reconfiguration with polybot. *IEEE/ASME Transactions on Mechatronics*, 7(4):442–451, 2002.

[35] E. Yoshida and A.and Tomita K.and Kurokawa H.and Kokaji S. Murata, S.and Kamimura. Evolutionary synthesis of dynamic motion and reconfiguration process for a modular robot m-tran. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 2, pages 1004–1010, July 2003.

[36] E. Yoshida, S. Murata, A. Kamimura, and H.and Kokaji S. Tomita, K.and Kurokawa. A self-reconfigurable modular robot: Reconfiguration planning and experiments. *The International Journal of Robotics Research*, 21(10-11):903–915, October–November 2002.

[37] Y. Zhan, M. Yim, and D.and Roufas K. Eldershaw, C.and Duff. Scalable and reconfigurable configurations and locomotion gaits for chain-type modular reconfigurable robots. In *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation,*, volume 2, pages 893–899, 2003.

[38] Y. Zhang and M. Roufas, K.D.and Yim. Software architecture for modular self-reconfigurable robots. In *Proceedings of the 2001 IEEE/RSJ International Conference on Robotics and Automation*, volume 4, pages 2355–2360, 2001.

A. Tables

A.1. Calibration values of the current modules

module - no.	zero-position [ms]	factor (for FPGA)
0	1480	90
1	1480	95
2	1550	96
3	1500	97
4	1580	97
5	1570	98

Table A.1.: Calibration of current servos used in the modules

A.2. List of RC-servo traders

company	homepage
Höllein	www.hoellein.com
Fliegerland Shop	www.fliegerland-shop.de
Conrad Electronic GmbH	www.conrad.de
Beck-Modellbau	www.modellbau-beck.de
KO PROPO USA, Inc.	www.kopropo.com
Modellbau Bichler	www.modellbau-bichler.de
Otto Modellbau	www.a1-modellbau.de
HOPE-Modellbau AG	www.hopmodell.ch
MODELCO SA	www.modelco.ch
FMA Direct	www.fmadirect.com

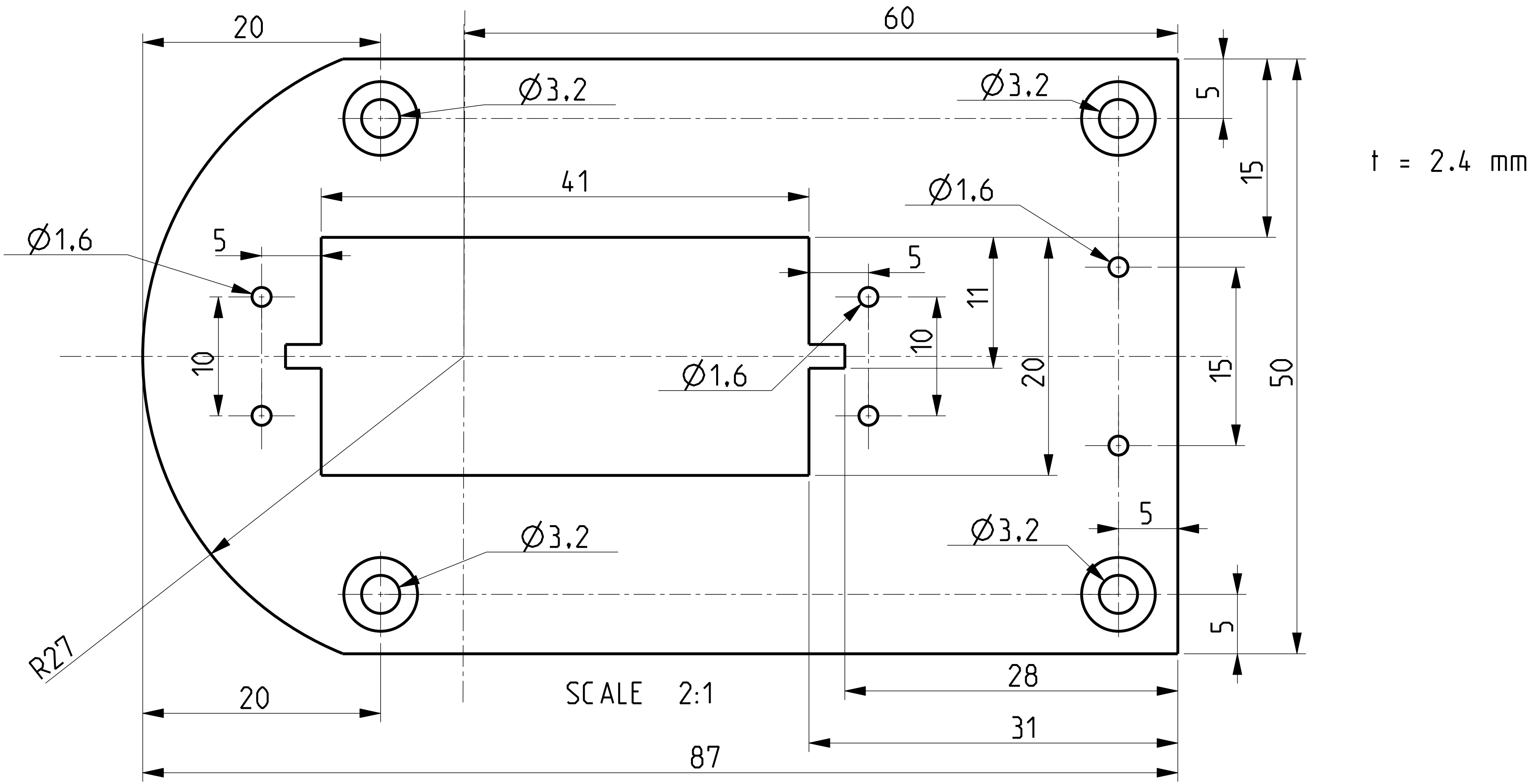
Table A.2.: List of servo brands and traders

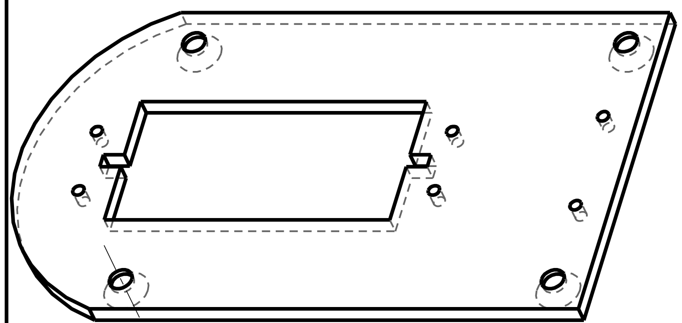
A.3. Bill of materials per module

No.	part	quantity	misc.
1	lower-layer	1	PCB, ProE: lower-layer.prt
2	upper-layer	1	PCB, ProE: upper-layer.prt
3	closure disc right	1	PCB, ProE: plaque-side-right.prt
4	closure disc left	1	PCB, ProE: plaque-side-left.prt
5	closure disc rear	1	PCB, ProE: plaque-tail.prt
6	servo plate	1	PCB; ProE: servo-plate.prt
7	fastener for rear disc	2	ALU, ProE: plaque-fix-tail.prt
8	U-profile	1	ALU, ProE: u-profile-1.prt
9	SERVO-S71 J/R	1	Conrad Electronic GmbH, order-no.: 224456
10	bronze bushing	1	Conrad Electronic GmbH, order-no.:237027
11	ALU-servo-lever	1	Conrad Electronic GmbH, order-no.:226636
12	Polyamid distance bolts 40 x M3	4	www.vogt.ch, order-no.:M30040.00.PA
13	axle 3mm	1	
14	machine screws M2 x 6 mm	6	
15	machine screws M2 x 8 mm	4	
16	machine screws M2 x 4 mm	2	countersunk
17	machine screws M3 x 8 mm	8	countersunk
18	machine screw M3 x 6 mm	2	countersunk
19	machine screw M2.5 x 4 mm	4	countersunk
20	Velcro fastener		
21	machine screw M3 x 8 mm	1	hexagon socket head

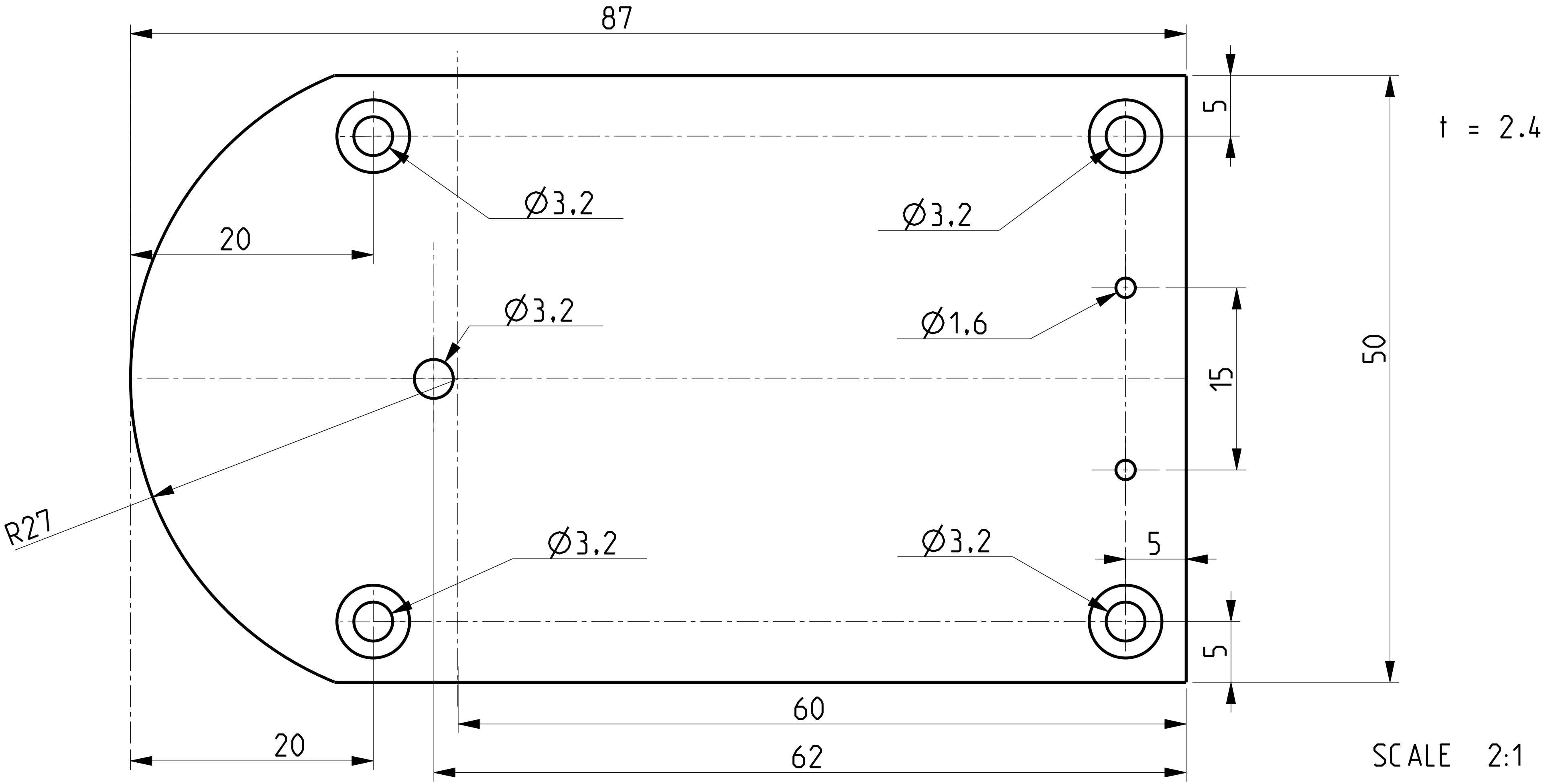
Table A.3.: Bill of materials

B. Drawings (CAD)

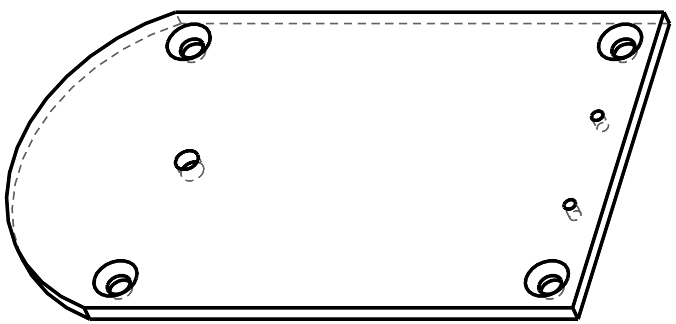




Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques		
Mod		Mod		Dessine	ditrich	Echelle 1:1
				Controle		
				Conf normes		
				Bon execution		
Sans nomenclature separee				N° commande		
Nomenclature sep de meme N°				Origine	Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° ident			Remplace		
EPFL	LOWER-LAYER				N° de dessin	



SCALE 2:1



Pos. | Quantite | Unite | N° d'identification | Denomination / Caracteristiques

Mod | Mod

Dessine: E.Dittrich
Controle
Conf normes
Bon execution
N° commande
Origine
Remplace

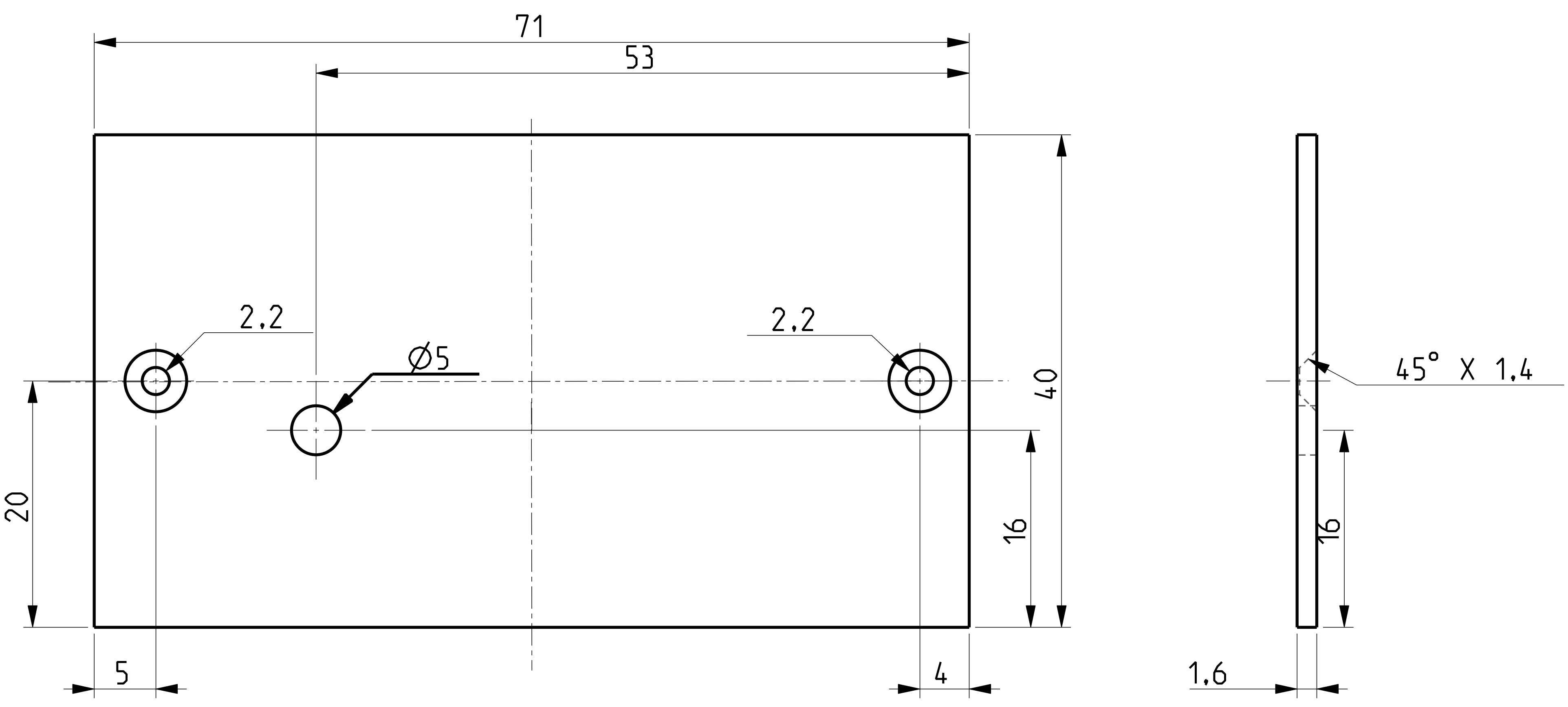
Echelle: 1:1

Sans nomenclature separee
Nomenclature sep de meme N°
Nomenclature sep de N° different | N° ident

Nb feuilles: 1
Feuille N°: 1
N° de dessin

EPFL

UPPER-LAYER



SCALE 2:1

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	
Mod		Mod		Dessine	E. Dittrich
				Controle	
				Conf normes	
				Bon execution	
Sans nomenclature separee				N° commande	
Nomenclature sep de meme N°				Origine	
Nomenclature sep de N° different	N° ident			Remplace	

Echelle 1:1

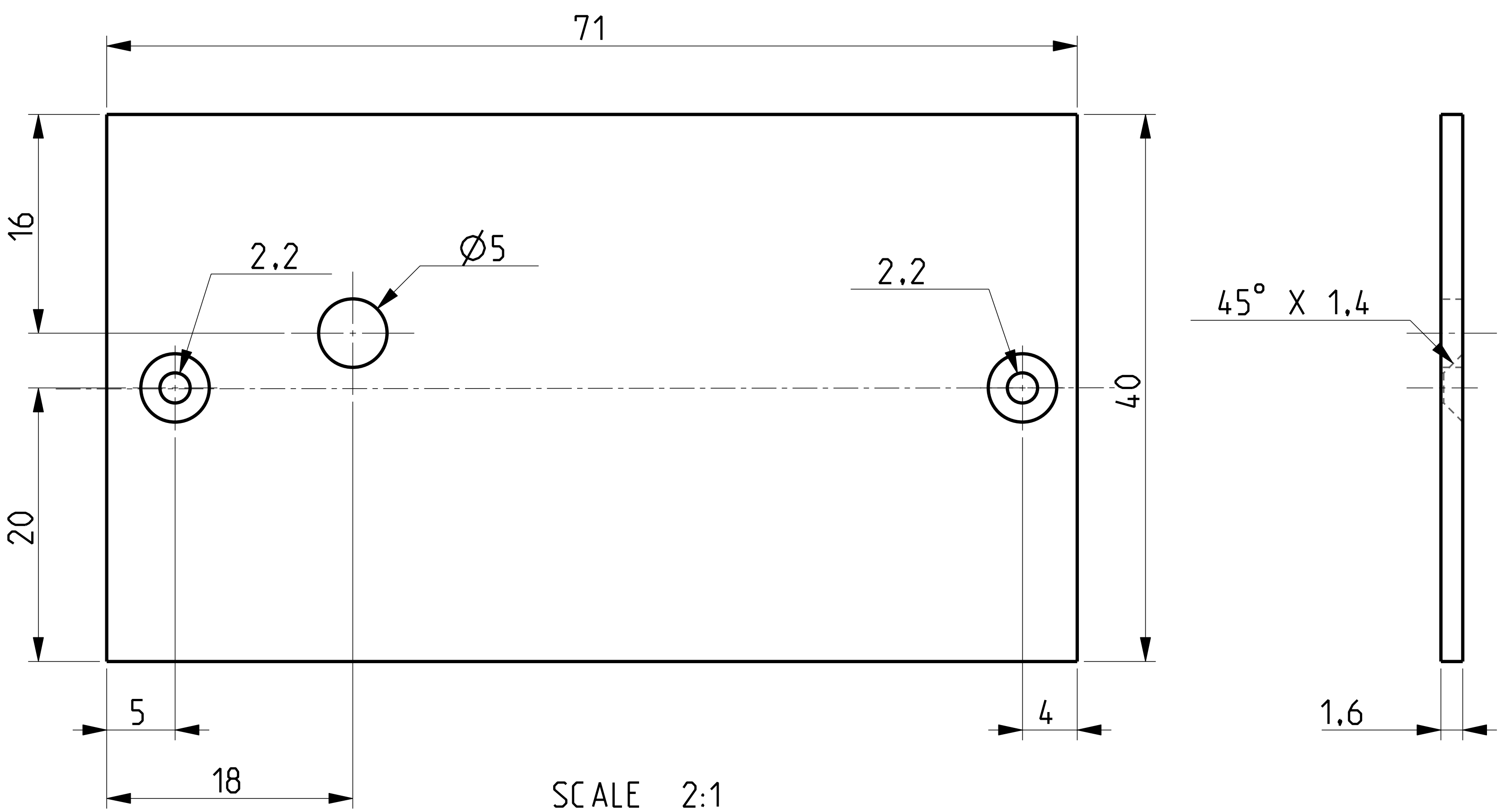
Nb feuilles 1

Feuille N° 1

N° de dessin

EPFL

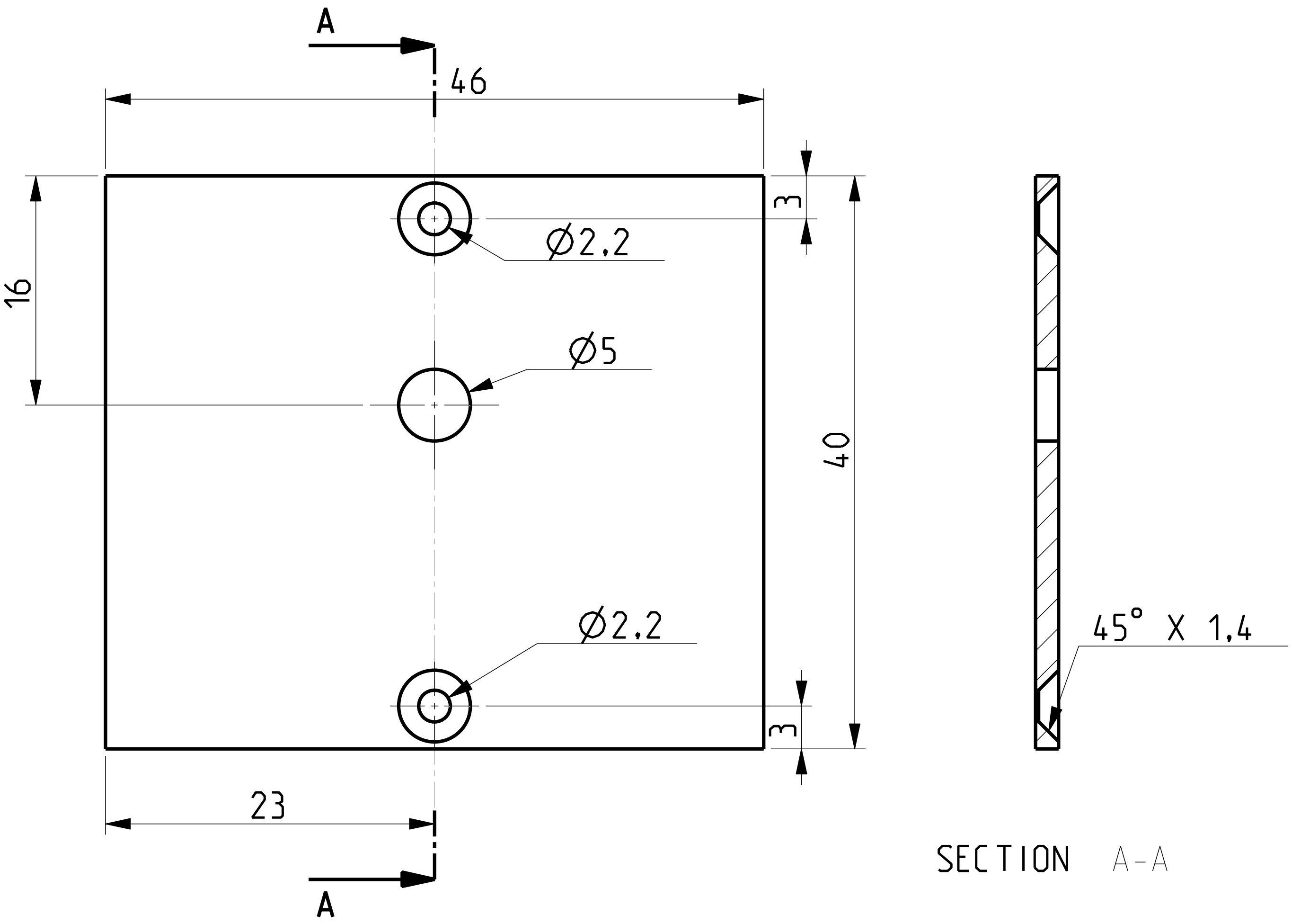
PLAQU-SIDE-RIGHT



SCALE 2:1

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques

Mod	Mod	Dessine	E. Dittrich	Echelle 1:1
		Controle		
		Conf normes		
		Bon execution		
Sans nomenclature separee		N° commande		
Nomenclature sep de meme N°		Origine		Nb feuilles 1 / Feuille N° 1
Nomenclature sep de N° different	N° ident	Remplace		
	PLAQUE-SIDE-LEFT			N° de dessin



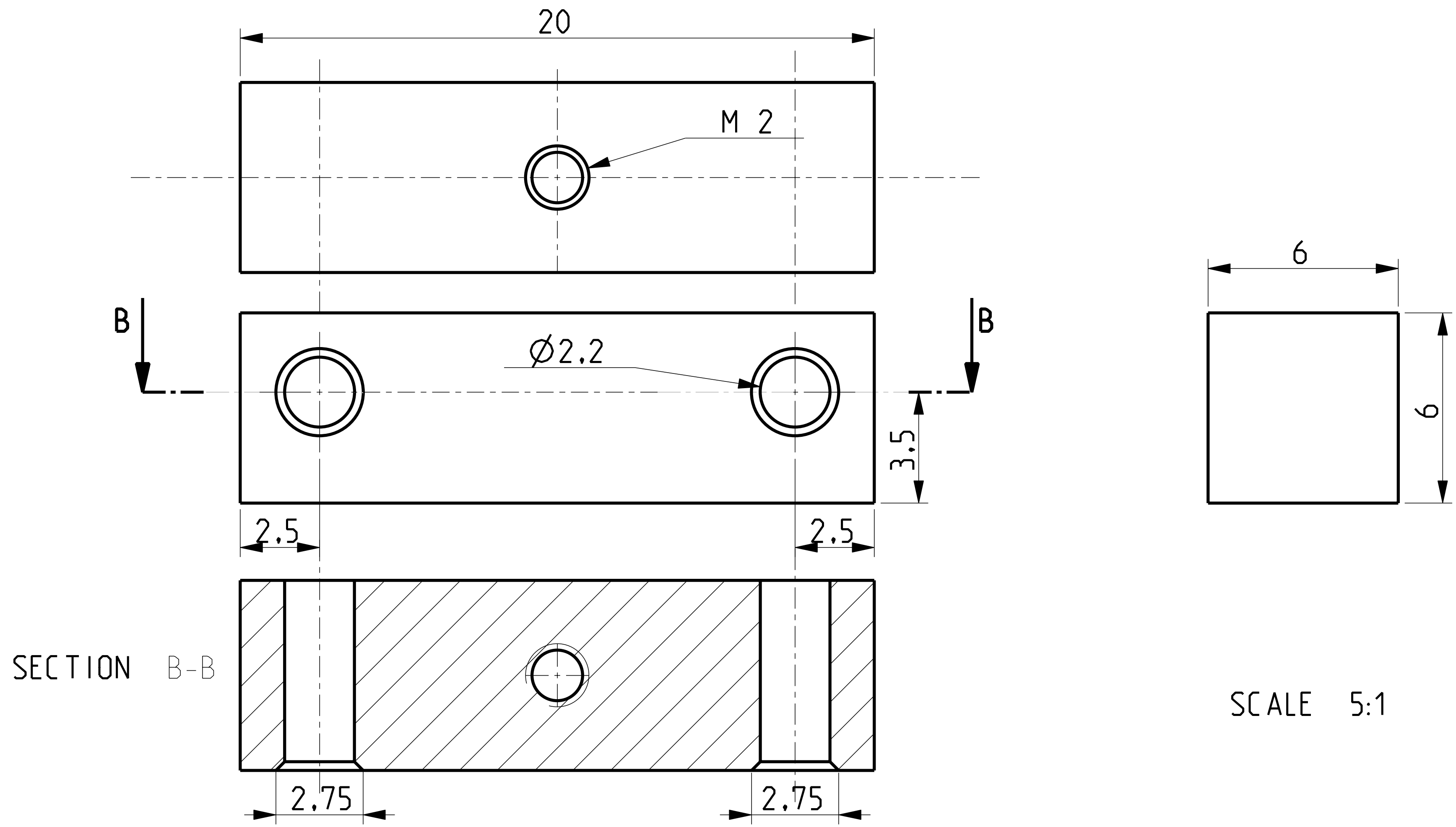
SECTION A-A

SCALE 2:1

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	

Mod	Mod			Echelle
		Dessine	E. Dittrich	1:1
		Controle		
		Conf normes		
		Bon execution		

Sans nomenclature separee		N° commande			
Nomenclature sep de meme N°		Origine		Nb feuilles	Feuille N°
Nomenclature sep de N° different	N° ident	Remplace		1	1
EPFL	PLAQUE-TAIL			N° de dessin	



Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	
Mod		Mod		Dessine	E.Dittrich
				Controle	
				Conf normes	
				Bon execution	
Sans nomenclature separee				N° commande	
Nomenclature sep de meme N°				Origine	
Nomenclature sep de N° different	N° ident			Remplace	

Echelle 1:1

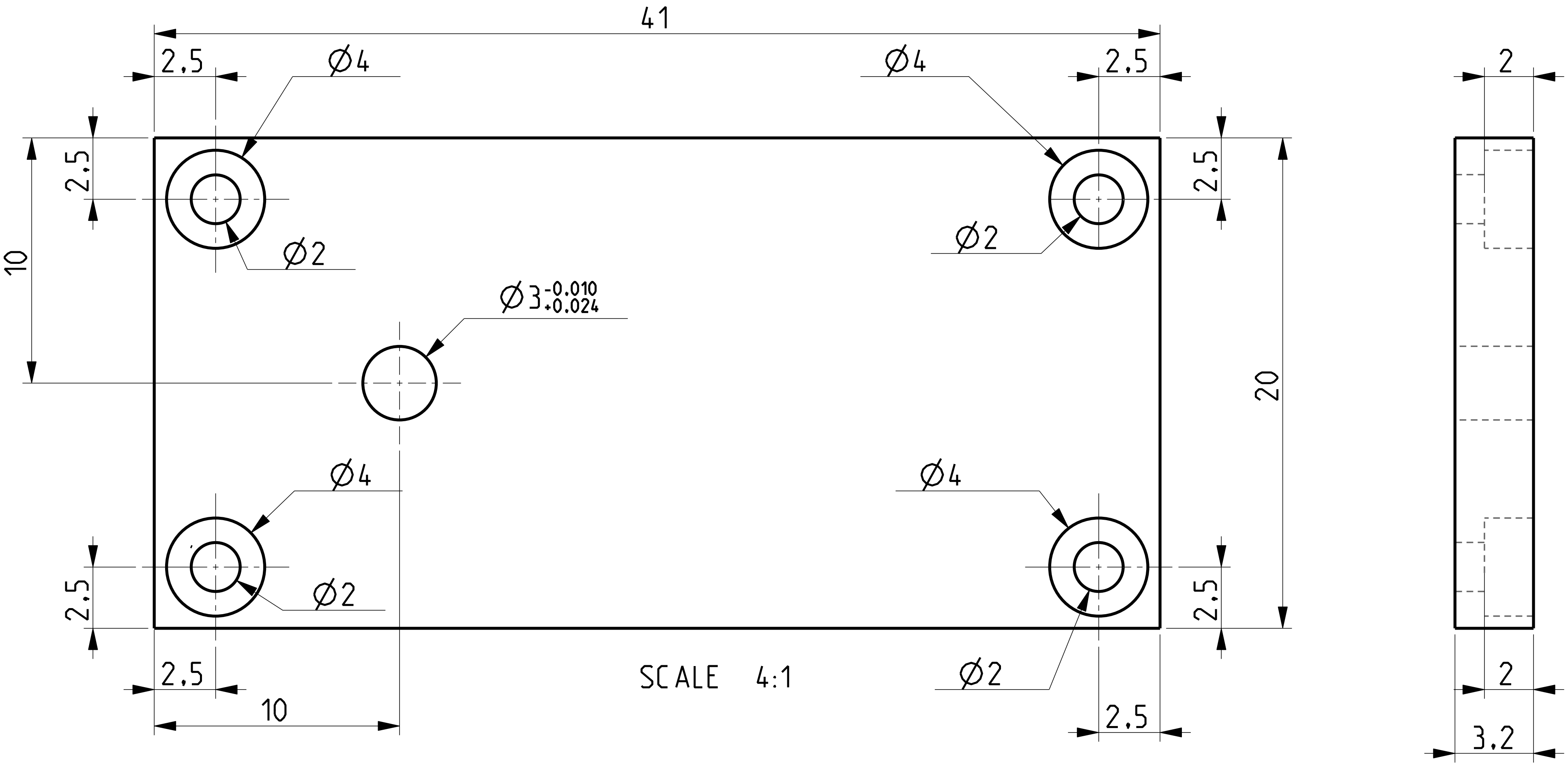
Nb feuilles 1

Feuille N° 1

N° de dessin

EPFL

PLAQUE-FIX-TAIL



Uebermasspassung (Einheitswelle) h6/R8

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	
Mod		Mod		Dessine	E. Dittrich
				Controle	
				Conf normes	
				Bon execution	
Sans nomenclature separee				N° commande	
Nomenclature sep de meme N°				Origine	
Nomenclature sep de N° different	N° ident			Remplace	

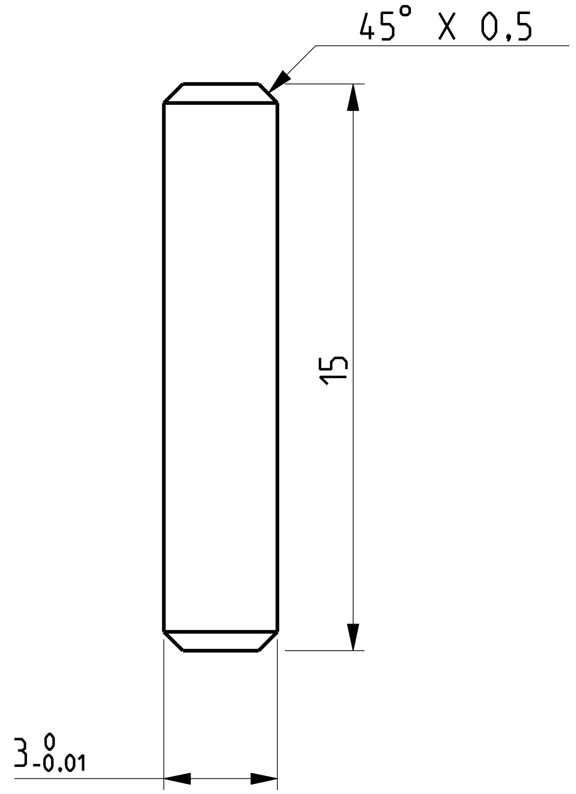
Echelle 1:1

Nb feuilles 1 | Feuille N° 1

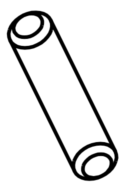
N° de dessin

EPFL

SERVO-PLATE



SCALE 5:1



SCALE 2:1

Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	
Mod		Mod		Dessine	E. Dittrich
				Controle	
				Conf normes	
				Bon execution	
Sans nomenclature separee				N° commande	
Nomenclature sep de meme N°				Origine	
Nomenclature sep de N° different	N° ident			Remplace	

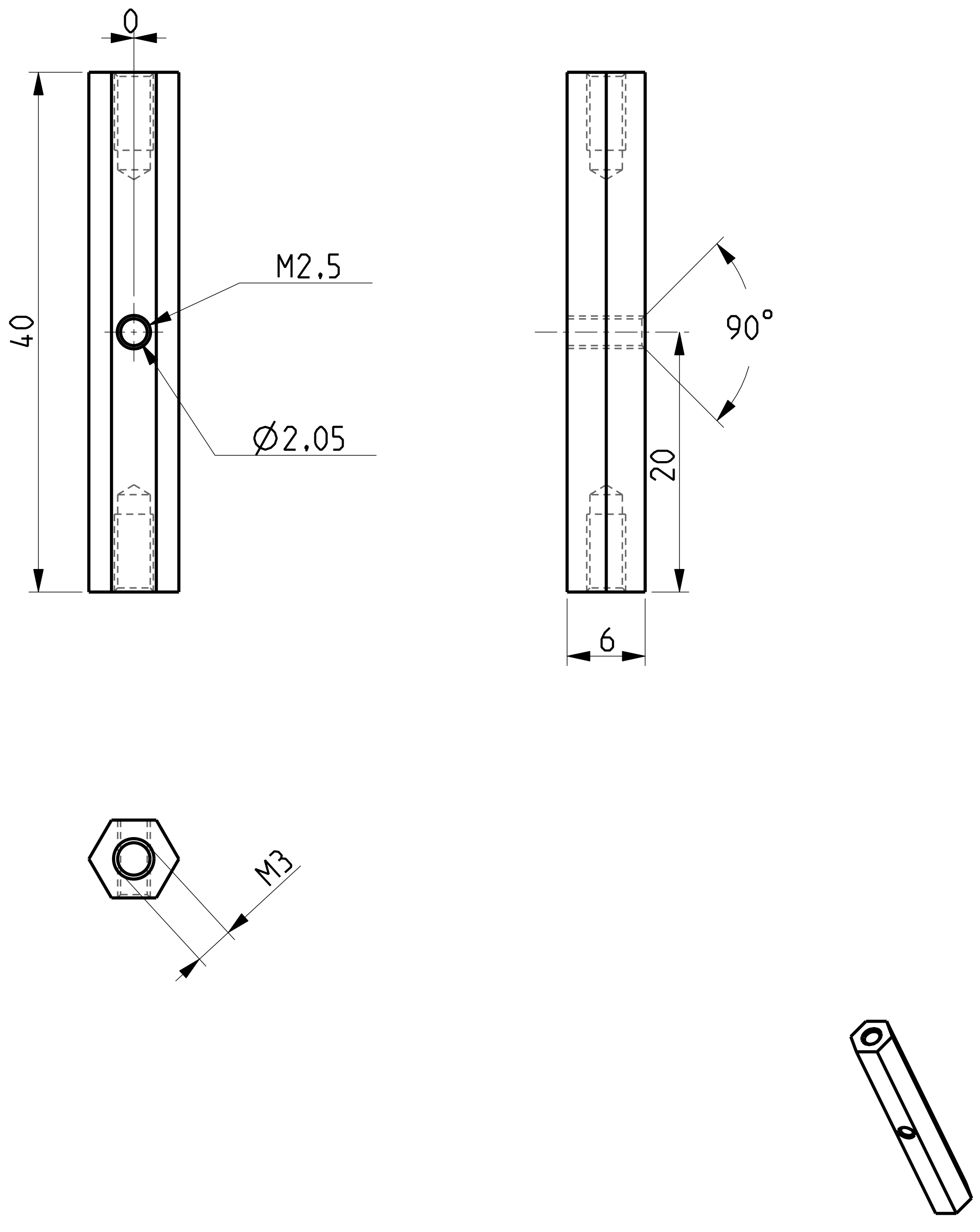
Echelle 1:1

Nb feuilles 1

Feuille N° 1

N° de dessin

PIN



SCALE 2:1

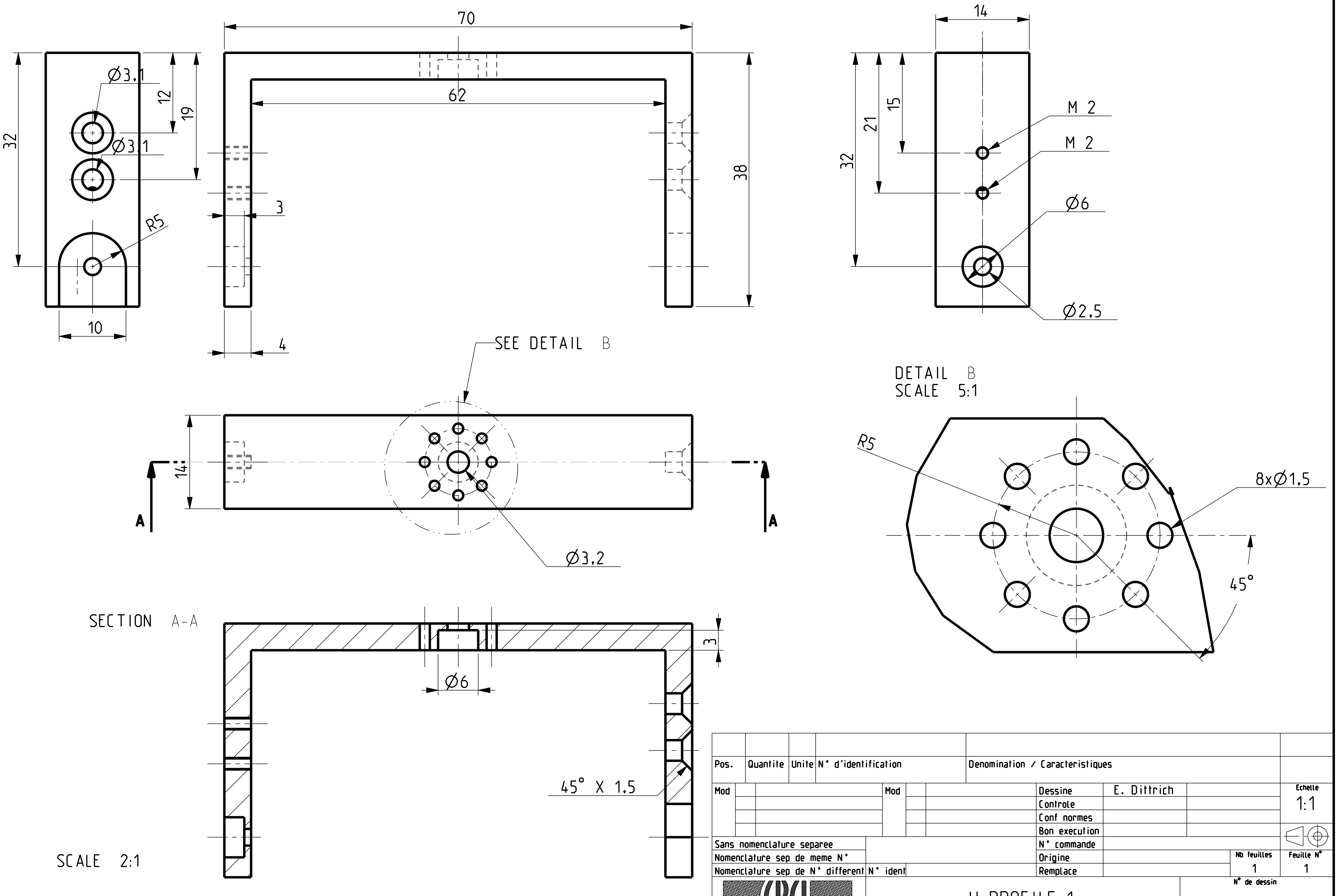
Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	

Mod			Mod			Dessine	E. Dittrich	Echelle 1:1
						Controle		
						Conf normes		
						Bon execution		
Sans nomenclature separee						N° commande		
Nomenclature sep de meme N°						Origine	Nb feuilles 1	Feuille N° 1
Nomenclature sep de N° different	N° ident					Remplace		

EPFL

SPACER-INT-THREAD

N° de dessin



Pos.	Quantite	Unite	N° d'identification	Denomination / Caracteristiques	
Mod		Mod		Dessine	E. Dittrich
				Controle	
				Conf normes	
				Bon execution	
Sans nomenclature separee				N° commande	
Nomenclature sep de meme N°				Origine	
Nomenclature sep de N° different	N° ident			Remplace	

Echelle 1:1

Nb feuilles 1 | Feuille N° 1

N° de dessin

EPFL

U-PROFILE-1