The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Odense, Denmark

Master Thesis in Robotics

The Odin Modular Robot: Electronics and Communication

Ricardo Franco Mendoza Garcia

Supervisor:
Kasper Stoy

Project Period: Autumn 2007
Project Due: February 29th, 2008

Ricardo Franco Mendoza Garcia (041079) - franco@mmmi.sdu.dk

Acknowledgements

Special thanks to the Modular Robotics Group at the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, which adopted me as a member of their innovative team and that everyday gives me much more than academical challenges.

I would also like to thanks the company Universal Robots Aps. which opened the doors of my career and provided me with economical support for most of the time this project lasted.

And, finally, I would like to thanks my family, my friends here in Denmark and there in Chile, and specially to Magdalena Dobrajska who is the main supporter of my dreams.

Abstract

This thesis describes the development of the electronics and communication system of the Odin robot, which is the first realization of the concept of *hierarchical robots*. A hierarchical robot is a robot made of modules which provide one or few functionalities, such as: power, actuation, sensing or structure. Consequently, a hierarchical robot is also a modular and a heterogeneous robot. Odin is made of two types of modules: links and joints. The links are implemented in different ways each to provide a single functionality, and the joints forward power and communication lines between neighbour links.

We develop the electronics for Odin in a layout that separates electronic components into two PCBs: the General and Specific boards. While the General board hosts processing power and it is embedded in every link of Odin, the Specific board hosts specialized hardware (e.g., battery monitor or motor driver) and it is embedded in specific links according to their functionality. We find this layout suitable for heterogeneous robots, since it forces to some extent the homogeneity of the electronics even though we are designing modules with different functionality.

We also develop a communication system which provides a flexible network topology made of local buses that can be extended by software-controlled switches, which we find suitable for modular robots as it improves transfer rates, space requirements and energy efficiency. For this hybrid communication approach we also visualize high level control advantages which we further probe as feasible with an analysis made on the efficiency of local and global communication in modular robots. For comparing the efficiency of local and global communication we further develop a local communication model for modular robots, which is useful for taking design decision and for elaborating control strategies on robots implementing communication systems as the one here proposed.

Contents

Chapter 1

Introduction

1.1 Modular Robots

Modular robots are robots built from modules. Single modules have limited functionality but many of them combined into larger structures allow more complex behaviours. The advantages of modular robots over fixed topology robots are: versatility, since they can be assembled into task suitable structures; robustness, since they are made of many replaceable modules; and cheapness, since their few types of modules *could* be easily mass produced [1].

Modular robots are classified according to different aspects of their mechanical design. One important parameter of classification is their homogeneity. Homogeneous modular robots are systems with just one type of modules, while heterogeneous modular robots are systems with two or more types of modules [2].

Another important parameter of classification of modular robots is the topology they form. Chain-type robots have modules that attach and detach in a string or tree topology, while lattice-type robots arrange modules in a lattice (or 3D virtual grid) occupying only discrete positions in space. Some modular robots can also form hybrid topologies [3, 4].

In both cases, if the modules of a robot are prepared to rearrange themselves into different shapes, the system is also called self-reconfigurable modular robot [5]. Self-reconfiguration is thought to provide adaptability to modular robots.

Considering that current achievements of modular robots are locomotion gaits across pre-defined terrains and self-reconfiguration sequences between pre-defined shapes and by following pre-defined steps [6, 7, 8], modular robots are still not prepared to perform real-world applications.

1.2 Motivation

At present, the complexity of modular robots conflicts with the promise of cheapness. Long development periods and expensive components are both required for building modules which have many integrated functionalities [2, 9]. From those functionalities, connectors prepared for self-reconfiguration is considered one of the most difficult aspect of design [2, 4, 10].

Even though it has been a while since the first modular robot, CEBOT, appeared back in 1998 [11], modular robots have not yet reached practical applications [5]. The last obeys the trend dictated by research which seems to be focused to a great extent on self-reconfiguration [12]. Most modular robots are self-reconfigurable and the most typical achievements include self-reconfiguration itself [2, 13].

Finally, self-reconfiguration may or may not be the best way to achieve adaptability. On their seek for adaptability, modular robots are strongly influenced by nature, but again many adaptable organisms do not perform self-reconfiguration (e.g., animals). In the same line, nature also exploits deformability when looking for adaptability [14], and bringing the concept of deformability into the field of robotics seems to produce simpler mechanisms than self-reconfiguration [13]. For us, humans, it would certainly be easier to take apart our legs when travelling in those uncomfortable plane seats, but we deform our bodies to fit in such a limited space instead. Anyway, we still adapt.

1.3 Goal

We want to investigate a new approach for building modular robots which attempts to simplify the complexity of modules, by implementing just one or few basic functionalities at these lower levels of the hierarchy. By basic functionalities we mean, for instance: power, actuation, sensing or structure. The idea is then to implement more advanced functionalities at higher levels of the hierarchy. In other words, we want to take the concept of modularity to functional levels. We classify systems that follows this new design approach as: *Hierarchical Robots* [9].

Our hypothesis is that systems made of simple, heterogeneous and monofunctional modules, which can then be combined into more complex metamodules, will shorten development time, simplify complexity, and, overall, reduce the cost of modular robots.

We also believe that hierarchical robots may encourage research focused on practical applications other than self-reconfiguration, as this functionality would be explored just if necessary.

Finally, we believe that hierarchical robots may achieve practical results faster than conventional modular robots, as their *development* is adaptable

to the direction of research. For instance, if a rotational actuator is thought to be more appropriate for research than a linear one, that rotational module could be easily developed without loosing previous designs, such as: power and sensor modules.

1.4 Approach

We build a preliminary realization of the concept of hierarchical robots: The Odin Modular Robot [13, 9]. This system is meant to be a heterogeneous modular robot whose modules are mono-functional. Each module could provide either: power, actuation, sensing, structure or any functionality demanded by research.

In addition, the Odin robot is a lattice-type deformable robot, made of joints and spring-loaded links, which also provides a framework to explore the advantages and disadvantages of deformability and prestressed structures [14, 13]. Thus, the system does not include connectors prepared for self-reconfiguration.

This work put special emphasis on describing requirements, design, implementation and testing processes of the electronics and communication system of Odin, as well as on discussing the advantages and disadvantages of their implementations.

1.5 Thesis Outline

We begin our work with Chapter 2, which reviews the electronics and communication systems of modular robots around the globe and provides the appropriate background for setting requirements and making design decisions in Chapter 3, which describes the different processes involved in the development of the electronics of Odin. Chapter 2 also provides the appropriate background for Chapter 4, which complements the review and describes the different processes involved in the development of the communication system of Odin. Then, Chapter 5 evaluates analytically the efficiency of local and global communication modalities in modular robots for discovering potential advantages of our proposed communication system and, finally, Chapter 6 concludes this work with the implications of the electronics and communication system developed along previous chapters.

Chapter 2

Related Work

Before working on the electronics of Odin, we look at different proposals of modular robots available around the globe which offer a logical and appropriate start point. We are conscious that the single functionalities we want to implement on our modules are already present in nowadays modular robots' design, all integrated though. In this chapter we describe hardware details of modular robots, whose authors have made clear publications about their electronics and communication system.

2.1 PolyBot (2002)

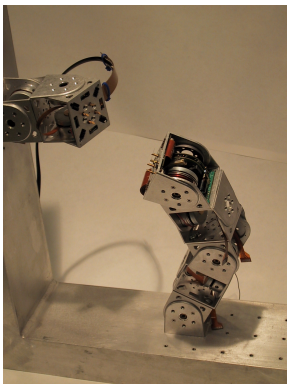
The PolyBot is the modular robot proposal of Palo Alto Research Center (PARC). This robot has demonstrated many modes of locomotion and within its accomplishments are the most active modules in connected system (56 modules) and the strongest actuation (5 modules cantilever). The PolyBot is a heterogeneous modular chain-type self-reconfigurable robot, which is made of two kinds of modules: segment and node [15].

The segments of PolyBot, version G3 (generation 3), have 1 rotational degree of freedom (DOF) and 2 connection surfaces. The nodes of PolyBot, version G3, are rigid with no internal DOF and have 6 connection surfaces [15]. Two connection surfaces of different modules can be docked in 90 degree increments, so the axes of two subsequent segment modules can be arranged in parallel or perpendicular. The PolyBot modules, version G3, are 50x50x50mm cubes (see Fig. 2.1).

2.1.1 Electronics

Each PolyBot module (segment or node) has a 32-bit 40MHz Motorola PowerPC 555 (MPC555) embedded processor with 1 megabyte of external RAM. Power and communication signals are transmitted through the connection surfaces, and the communication system is implemented over a CANBus.

(a) A PolyBot module.



(b) A set of PolyBot modules.

Figure 2.1: The PolyBot modular robot is a heterogeneous modular chain-type system, made of two kinds of modules: segment and node. Fig. (a) shows one segment module and Fig. (b) shows a worm-like entity made of several segment modules. The modules can be docked in 90 degree intervals through connection plates (surfaces).

The modules are using as many CAN controllers as connection surfaces they have: 2 processor-embedded controllers in the segment modules and 6 external controllers in the node modules. The segment modules communicate over a global CANBus with up to 1Mbps. The node modules have switching and routing capabilities, and they can bridge several buses of segment modules into more global mediums [16].

Each segment module is equipped with a brushless DC (BLDC) motor for actuation, hall sensors for measuring commutation and joint position (0.43 degree resolution), 1 joint angle potentiometer, and 2 accelerometers for measuring orientation relative to gravity [17].

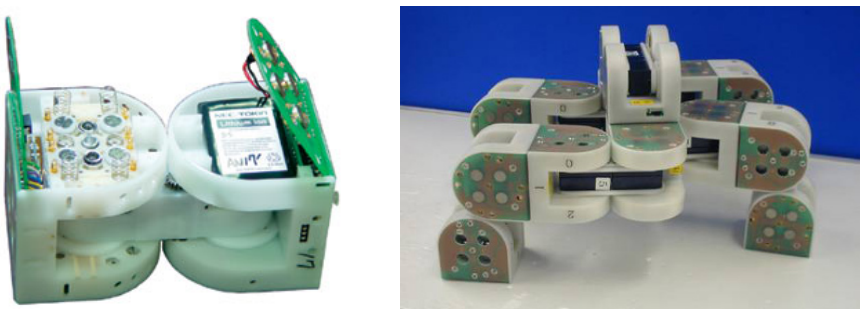
2.1.2 Observations

The processing power and external RAM have not yet been completely used, the node modules add delay to messages being transmitted between different buses, and accelerometers are useful at performing locomotion gait selection and specialized gaits. Further, the system is currently run tethered to a power supply, and the increment of the voltage been distributed to the modules made the robot more stable. The communication being performed is global.

2.2 M-Tran II (2002)

The M-Tran II (generation II) is the modular robot proposal of the National Institute of Advanced Industrial Science and Technology of Japan (AIST) and the Tokyo Institute of Technology (TiTech). Its name stands for Modular Transformer and one of its accomplishments is the most robust self-reconfiguration (14 non-repeating attach/detach steps). The M-Tran II is a homogeneous modular self-reconfigurable robot, with a hybrid architecture between chain and lattice topologies [3].

The modules of M-Tran II has 2 rotational DOFs and 6 connection surfaces. They are made of two semi-cylindrical parts and a link, making a total dimension of 60x120x60mm. The semi-cylindrical parts are called passive and active, based on their roles when docking neighbours. Each half hosts 3 connection surfaces (see Fig. 2.2).



(a) A M-Tran II module. (b) A set of M-Tran modules.

Figure 2.2: The M-Tran II modular robot is a homogeneous system with a hybrid topology (chain and lattice). Fig. (a) shows the only type of module in M-Tran II system and Fig. (b) shows a four-legged walker made of several modules of M-Tran I (generation I). M-Tran I follows the same mechanical principles of M-Tran II.

2.2.1 Electronics

Each M-Tran II module has an 8-bit 20MHz Neuron Chip (master) and three 8-bit 20MHz PICs (slaves) microcontrollers [3]. Power, global communication and local communication signals are passed through the connection surfaces, and the communication mediums are wired. Global communication connects all the modules of M-Tran II with one external host PC, which is part of the control scheme of the system.

The communication between the 4 internal microcontrollers is established over a 4800bps asynchronous serial line, the global communication is established over a 39Kbps RS485 bus (under LON higher level protocol), and the local communication between neighbours is established over a 4800bps

serial line. The transfer rate of the global communication is restricted to 39Kbps due to the exclusion of termination impedance in the bus.

The internal and global communication are managed by the master microcontroller, while local communication is managed by 2 slaves. The passive part hosts the master and one slave for control, power supply and communication; the active part hosts one slave for docking and communication; and the link hosts one slave for controlling motors.

The links of the M-Tran II modules embed two geared DC motors and two motor driver chips for rotating the passive and active parts. Each motor produces 19.8Kg-cm of torque, which is enough to lift-up two other modules. The torque of the motors is regulated by PID control algorithms. Further, the passive part has accelerometers for sensing the orientation of the modules.

There are two ways of supplying power to the system: by using an external tether connected to one of the modules (8V to 20V) or by using internal 700mAh 3.8V lithium-ion rechargeable batteries (we assume a pack of them connected in series). The internally regulated voltages used in the M-Tran II modules are 5V and 8V.

2.2.2 Observations

Currently, there is a newer version of the M-Tran modules, M-Tran III (generation III), which includes several improvements over the described hardware. Nevertheless, as we do not have detailed specifications of M-Tran III, we decided to document the previous version.

At M-Tran III, the master CPU is upgraded to the 32-bit 100MHz SH-2 microcontroller from Renesas Technologies, the slaves CPUs are changed to H8 microcontrollers also from Renesas Technologies, the global communication bus is upgraded to CANBus, and the internal batteries are changed to lithium-polymer.

M-Tran II uses the external powering tether for connecting the host PC computer to the modular system. Thus, the tether also includes communication lines. The transfer rate of the communication between microcontrollers (4800bps) may be a bottleneck, as the transfer rates of the local and global communication are the same or higher.

The authors said that for locomotion along a rough terrain or stairs, the system needs more sensors to interact with the environment (e.g. touch sensor or force sensor). Also, they mention the system must become more intelligent and autonomous to select appropriate configuration and motion by itself [7].

The power of the modules is consumed to a great extent by the self-reconfiguration process, and the system can perform global and local communication.

2.3 Atron (2004)

The Atron is our first modular robot proposal here at the University of Southern Denmark (USD) and its design is guided by the considerations on how to reduce control complexity of self-reconfiguration. The Atron is a homogeneous modular self-reconfigurable robot, complying with a surface-centred cubic lattice structure [18].

The Atron modules are made of two hemispheres joint together by a rotation mechanism which provides the system one DOF. Each hemisphere has two passive female connectors and two active male connectors which are able to hook complementary female connectors on neighbouring modules. In addition, the hemispheres rotate around the equator line by unlimited amounts of turns and a rotational locking mechanism is able to hold a relative position of the hemispheres every 90 degrees intervals. The Atron modules form a sphere with diameter of 114mm (see Fig. 2.3).

(a) An Atron module. (b) A set of Atron modules.

Figure 2.3: The Atron modular robot is a homogeneous modular lattice-type system, complying with a surface-centred cubic lattice structure. Fig. (a) shows the only type of module in the Atron system and Fig. (b) shows two entities made of several modules: a worm-like crawler (left) and a wheeled car (right).

2.3.1 Electronics

Each hemisphere of the Atron modules has one 8-bit 1MHz ATmega8 and one 8-bit 16MHz ATmega128 microcontrollers. The modules are equipped with a slip-ring placed in the junction of the two hemispheres which forwards power and communication lines between halves. In addition, the mechanical connection established by the connectors provides an electrical link for power transmission between modules.

There is one IrDA transmitter/receiver pair around each connector, what establish four infrared neighbour-to-neighbour communication channels per

hemisphere. The last allows to communicate with the four adjacent modules the surface-centred cubic lattice allows to have per hemisphere.

The communication between the ATmega8 and the ATmega128 on each hemisphere is established over an I2C bus, the communication between the two ATmega128 microcontrollers located on separated hemispheres is established over a RS485 bus (passing across the slip-ring), and the communication between the ATmega128 microcontroller and IrDA devices on each hemisphere is established over a RS232 bus. The transfer rate of the infrared communication channels established between connectors of different modules is 9600bps.

At one hemisphere, the ATmega8 controls rotational mechanism, rotational lock and two active male connectors, and the same microcontroller reads tilt and encoder information; while the ATmega128 controls four IrDA communication channels and coordinates the whole functionality of the module. At the other hemisphere, the ATmega8 controls the power sharing and re-charging features; while the ATmega128 controls four IrDA communication channels and two active male connectors.

The Atron modules have four DC motors actuating four active male connectors each (two per hemisphere) and one BLDC motor, controlled by a dedicated driver chip, actuating the rotational mechanism. In addition, one solenoid actuates the rotational lock, one tilt sensor measures 2-planes inclination, one central encoder disk measures absolute rotational angle and angular velocity, and shunt resistors measure different currents sent to the DC motors. The IrDA devices are also used for simple distance measurements and eight LEDs are included for debugging purposes.

Each module is energized either by two internal serial-connected lithium-polymer batteries providing 920mAh at 7.4V [19] or by an external unregulated power line coming from neighbour modules or from an external power supply. This scheme allows the batteries of the modules to be re-charged by using a unique tether and the power consumption of the whole system to be balanced (the modules enable internal batteries just when the external voltage drops).

2.3.2 Observations

Not to forget, the Atron design was guided by the considerations on how to reduce the control complexity of self-reconfiguration. Power sharing is convenient to compensate the motion-active modules discharge by taking power from the motion-passive modules. Nevertheless, power sharing and rotational lock features are not working properly in the Atron.

The infrared communication channels create problems for directing communication towards nearest neighbours, since reflections on adjacent surfaces forward the infrared signal to more distant modules. In despite of that, this problem has been overcame by implementing communication filtering algo-

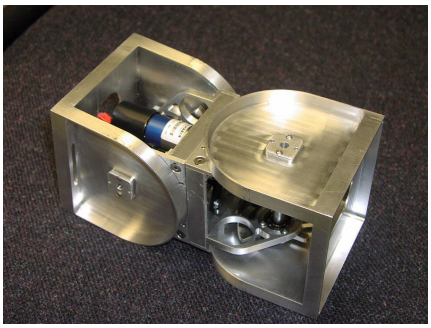
rithms, as described in [20].

We are currently working in a new iteration of the electronics of the Atron modules, in which we are trying to improve problematic features like the ones described above, and also trying to provide more flexibility to the hardware blocks by using FPGAs. The communication established with neighbouring modules is local.

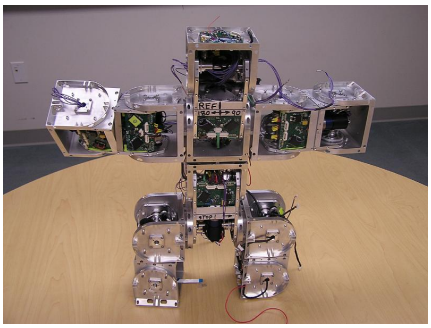
2.4 SuperBot (2006)

The SuperBot is the modular robot proposal of the University of Southern California (USC), which has been developed as a deployable self-reconfigurable robot for real-world applications [5]. Within the accomplishments of SuperBot are the robot with the longest distance run autonomously (750m with one charge) and the control approach with most behaviours based on topology[1] (3 behaviours). The SuperBot is a homogeneous modular self-reconfigurable robot, with a hybrid architecture between chain and lattice topologies.

The modules of SuperBot have 3 DOFs (roll, pitch, yaw) and 6 identical connection surfaces. The robot is built with self-reconfiguration in mind but the structures are still reconfigured by hand. The modules are made of two 84x84x84mm linked cubes, making a total length of 168mm. Each half hosts 3 connection surfaces (see Fig. 2.4).



(a) A SuperBot module.



(b) A set of SuperBot modules.

Figure 2.4: The SuperBot modular robot is a homogeneous system with a hybrid topology (chain and lattice). Fig. (a) shows the only type of module in the SuperBot system and Fig. (b) shows a humanoid entity made of several modules. The structures are still reconfigured by hand.

[1]Developed for the CONRO modular robot [21], the predecessor of SuperBot.

2.4.1 Electronics

Each half of a SuperBot module has an 8-bit 16MHz ATmega128 microcontroller and each microcontroller communicates with the 3 connection surfaces on each half. Power and communication signal are passed through the connection surfaces and the communication medium is infrared. There are 4 infrared receiver LEDs and 1 infrared transmitter LED per connection surface.

The communication between the microcontrollers is over a 400Kbps I2C bus, and the communication between a microcontroller and its 3 connection surfaces is over a 1Mbps SPI bus. The last allows to establish a 230KBaud RS232 line (IrDA) between connection surfaces of different modules (neighbour-to-neighbour communication). The infrared communication can be also used for guiding the docking process and sensing proximity of objects.

The modules use 3 MicroMo DC electric motors for producing 6.38N-m on 2 actuators and 1 rotating central part, and they also have embedded a 1600mAh 7.4V lithium-polymer battery pack and 1 accelerometer. One microcontroller operates 1 motor and the battery power management unit in one half (master), and the other microcontroller operates 2 motors in the other half (slave).

Due to a switching mechanism that allows the modules to sequentially spread the power to neighbours, all the internal batteries of the modules can be recharged by using a unique tether [4].

2.4.2 Observations

More degrees of freedom in a module may give more dexterity to the robot, and dexterity is a power saving feature when avoiding objects. Even if the communication is meant to be neighbour-to-neighbour, there can be reflexions to other modules before a complete docking. In spite of DC motors are not supposed to be the most efficient, the PolyBot is the most efficient modular robot at doing locomotion. The communication being performed is local.

2.5 Discussion

2.5.1 Processing Power

The electronics embedded in modular robots is rather complex. Most modular robots are multi-cpu systems providing many integrated features. The simplest modular robot is PolyBot and it is the only one that has a single cpu (a powerful one, though). Nevertheless, the common point to all the systems is that they have one cpu for controlling few functionalities. That

suggest us the implementation of a single-cpu electronic scheme in our modules, since they explicitly aim to provide few functionalities (ideally just one).

2.5.2 Power Supply

Since the idea of power sharing involve high currents circulating across modules, the voltage distributed within modular robots should be as high as possible (more than 7.4V in our reviews). The highest the voltage the lowest the current for a specific power requirement. Consequently, as the power supplies of Odin will be implemented in specialized modules (they will not be embedded in every module), we have to design our electronics with this advise on mind.

2.5.3 Actuation

From literature, BLDC motors seem to be the actuators with the highest output torque and efficiency. These two characteristics may be tempting when choosing the final mechanics for actuator modules. Nevertheless, the recognized advantages of BLDC motors depend to a great extent on the performance of the commutation algorithms. Ironically, the modular robot with the highest efficiency is SuperBot, which uses several easy-to-control DC motors for actuation. If selecting BLDC motors as actuators for Odin, we should use specialized drivers chips with the most advanced commutation algorithms (e.g., sinusoidal commutation) to fully exploit the advantages of this technology.

2.5.4 Sensing

As quoted in [12, 5], sensor information for modular robots provides many research opportunities and our review confirms that truth. Reviewed modular robots include typical feedback sensors attached to their actuator mechanisms, but they do not include many sensors oriented towards the interaction with the environment. The only sensors we could relate with that task are accelerometers and infrared channels implemented at the communication interfaces. Nevertheless, the infrared sensors are not originally thought to interact with the environment and, therefore, not optimal for this task. These facts give us freedom to explore different hardware proposals in this context.

2.5.5 Communication

We do not have a clear conclusions about what is the best communication approach within modular robots. As PolyBot uses global, M-Tran uses local and global, Atron uses local, and SuperBot uses local, there is

not design trend in this context. In addition, the most important achievements of M-Tran, the system having both communication approaches, are self-reconfiguration oriented, which still does not give us much information considering our research focus. Nevertheless, we did not report the control approaches taken by the different modular robots (since we think it goes beyond the scope of this work), but there we realized that control algorithms of modular robots are highly dependant on the chosen communication modality. Since it is interesting for us to have as much flexibility as possible, we should implement both communication approaches in our modules.

2.5.6 Others

Ignoring self-reconfiguration, PolyBot's modules are the best approximation to our idea of hierarchical modules. They are actuators performing only 1-DOF rotational action and they do not embed power supplies on-board (at least for now). On the other hand, SuperBot, the only reviewed modular robot that still lacks of self-reconfiguration capabilities, has shown very interesting achievements at performing practical tasks, what matches our current research concept.

Chapter 3

Electronics

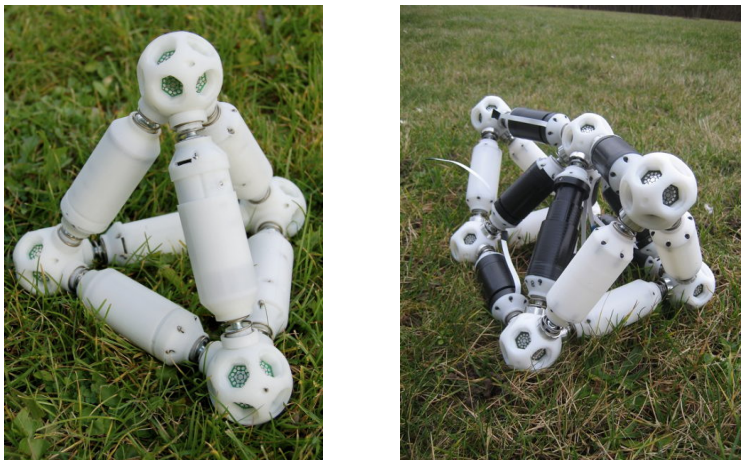
The Odin robot is a heterogeneous system made of two types of modules: links and joints. Links are mono-functional modules, which could provide either: power, actuation, sensing, structure or any functionality demanded by research. In this chapter we describe the electronics of the Odin robot, whose implementation comprises a General board with processing power which is embedded in every link, and a Specific board with specialized hardware (e.g., battery monitor or motor driver) which is embedded in specific links according to their functionality. We find this electronic layout convenient for heterogeneous robots, since it keeps to some extent the homogeneity of electronic components (i.e., lower production cost).

3.1 Introduction

The Odin robot is a preliminary realization of the concept of hierarchical robots, which is made of two types of modules: links and joints. The links are implemented in different ways each to provide one basic functionality, and the joints forward communication and power between neighbour links [13]. Fig. 3.1 shows two assemblies of the Odin robot.

As we are talking a heterogeneous system, we have to put special attention to the electronics of the modules. Even though electronics is not the most expensive component of modules, it could produce important secondary effects that do increase the cost, such as: complex assembly processes and too broad range of electronics components [1].

We begin this chapter with Section 3.2, which describes the mechanics of Odin or the physical framework for embedding our electronics. After that, Sections 3.3, 3.4 and 3.5 describe the requirements, design and implementation processes of the electronics and, then, Section 3.6 tests the implementation of such electronics with simple locomotion experiments. Finally, Sections 3.7 and 3.8 discuss the advantages and disadvantages of our implementations and the future direction of our work.



(a) Odin version 1. (b) Odin version 2.

Figure 3.1: The Odin robot, a preliminary realization of the concept of hierarchical robots, which is made of two types of modules: links and joints. The links are implemented in different ways each to provide one basic functionality, and the joints forward communication and power between neighbour links.

3.2 Mechanics of Odin

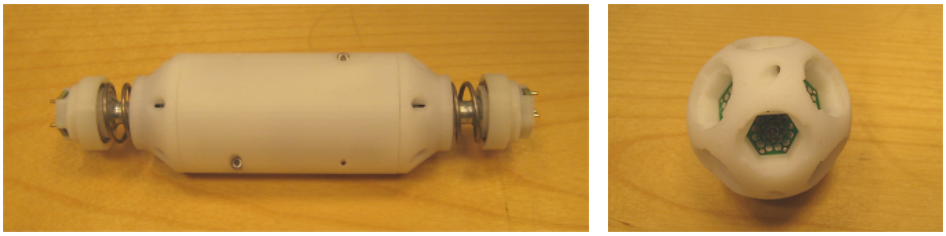
As mentioned before, Odin is a heterogeneous modular lattice-type robot made of two types of modules: links and joints. These modules can be locked together by a flexible connector-socket mechanism which is *not* prepared for self-reconfiguration but allows the system to perform movements by deforming its lattice [13].

The links are cylinders with 35mm diameter and 110mm length which are spring loaded at each extreme for assembling flexible structures; and the joints are balls with 50mm diameter which have embedded 12 sockets for hosting links [13]. The links and joints of Odin are shown in Fig. 3.2.

While the links can be implemented in different ways to provide either: power, actuation, sensing or structure functionalities; the joints arrange the links into specific lattices. In this case, the 12 sockets in the joints create the pattern for structures complying with the cubic closed packing lattice (CCP) [22]. Fig. 3.1 shows two assemblies of Odin.

3.3 Requirements

As we are aiming the Odin robot as a heterogeneous many-modules system, the electronics considered in the design has to be flexible enough to meet low production costs. By flexible electronics we mean that the components present at any link should be used as many times as possible along other



(a) A link of Odin. (b) A joint of Odin.

Figure 3.2: The two basic types of modules of the Odin robot. Fig. (a) shows a link of Odin which is spring loaded at each extreme for assembling flexible structures, and Fig. (b) shows a joint of Odin which embeds 12 sockets for hosting links. The links are cylinders with 35mm diameter and 110mm length, while the joints are balls with 50mm diameter.

links. Ideally, the same electronics should be used in power, actuator, sensor or structure links.

3.4 Design

3.4.1 Layout

We begin the hardware design with a high level description of the electronics. Our intention is to identify components which are common to all the links of the system (regardless of functionality).

Based on Chapter 2, we decided that the links of Odin will include a single CPU: as the links are thought to provide just one or few functionalities, no more processing power is required. Consequently, one microcontroller and associated electronics (e.g., low-level voltage regulator and decoupling capacitors) are the first components to include on every link of Odin.

Also based on Chapter 2, we decided that we are going to implement local communication, with the possibility of extending the scope of the local buses by using software-controllable switches. This peculiar hybrid communication approach [23], will be detailed in Chapter 4. In this way, the communication topology will be flexible and we will be able to implement different control algorithms. Thus, two communication chips, two software-controllable switches and associated electronics are other components to include on every link of Odin.

As there will be power links energizing many other links, we also decided that power lines will be forwarded through the connector-socket mechanism [13]; and, finally, we decided that the identified components will be distributed over two printed circuit boards (PCBs), from now on called: General and Specific boards. The reason for the last decision will be shown

next.

3.4.2 General Board

This board will host one microcontroller (CPU), one low-level voltage regulator, one communication chip (RXTX1), one software-controllable switch (Switch1) and associated electronics. We conveniently selected these components for having an identical General board embedded along all the links of Odin.

As there will be just one microcontroller per link, we have to send appropriate signals to the electronics placed at the Specific board. The signals selected are: UART port, PWM channels, SPI port, ADC channels and Digital I/O (maybe multiplexed); since they allow to acquire information from and/or to control a wide range of electronic devices.

3.4.3 Specific Board

This board will host one communication chip (RXTX2), one software-controllable switch (Switch2) and associated electronics. We did not include these components in the General board, on behalf of the communication topology.

As we imagine the placement of the General and Specific boards one at each extreme of a link (i.e., near to the local buses as shown in Chapter 4), we decided to place one communication chip on each board. Fig. 3.3 shows the location of the General and Specific boards within a link of Odin.

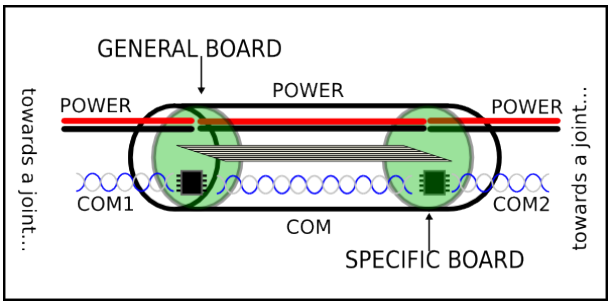


Figure 3.3: Location of the General and Specific boards within a link of Odin. The General and Specific boards host the embedded electronics of Odin and they are placed near to the extremes of a link.

In addition, the Specific board will also host components which are related to the specific functionality of a link. For instance, a power link may require electronics for charging batteries while an actuator link may require electronics for controlling a motor.

Thus, the Specific board will be a template to begin the hardware design with, whose implementation may change between links with different functionalities. Fig. 3.4 summarizes the layout of the electronics of Odin.

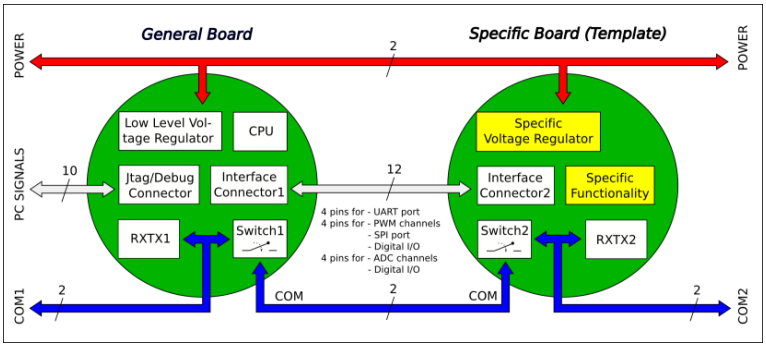


Figure 3.4: Layout of the electronics of Odin. The General board will host components which are common to every link of Odin. Therefore, the General board will be exactly the same along all the links of Odin. The Specific board will host some components which are common to every link of Odin (white blocks), but also components determined by the specific functionality of the link (yellow blocks). Thus, the Specific board may differ between links with different functionality.

3.5 Implementation

3.5.1 Layout

The General and Specific boards embedded on each link are connected by a flat cable with 12 microcontroller signals, a pair of power cables propagating the system voltage across the links and a serial communication bus for extending the scope of local buses (see Fig. 3.3).

The CPU selected for the Odin robot is a 32-bit 50MHz AT91SAM7S256 microcontroller from Atmel, which provides: high speed, low power requirements and plenty of peripherals in a very small footprint.

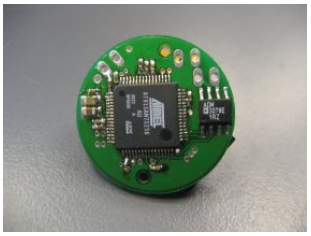
Considering that the links have 35mm diameter, the General and Specific boards are constrained to a circle with 25mm diameter. As a result, we manufactured 4-layered PCBs to obtain a high density of components.

3.5.2 General Board

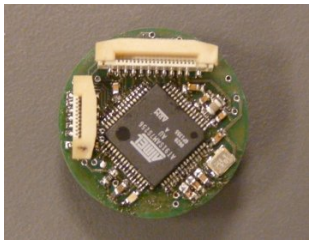
We made two implementations of the General board. At first we implemented the General board version 1 (V1) which operates from a system voltage of 5V. Nevertheless, as 5V voltage distributed across the system

translated into too high currents, we implemented the General board version 2 (V2) which operates from a system voltage of 11.1V instead. Fig. 3.5 shows the two implementations of the General board.

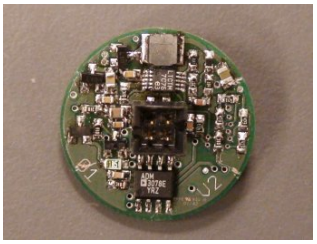
(a) General board V1 (front).



(b) General board V1 (back).



(c) General board V2 (front).

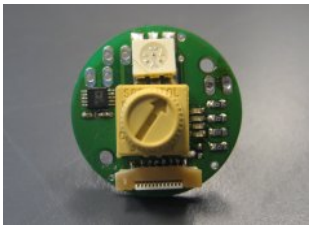


(d) General board V2 (back).

Figure 3.5: Implementations of the General board of the Odin robot. Fig. (a) and Fig. (b) show the front and back sides of the General board V1, which operates from a system voltage of 5V. Fig. (c) and Fig. (d) show the front and back sides of the General board V2, which operates from a system voltage of 11.1V. The General board is the same along all the links of Odin and it provides processing power. General board V2 is an improvement of General board V1. The General board is a 25mm diameter printed circuit board (PCB).

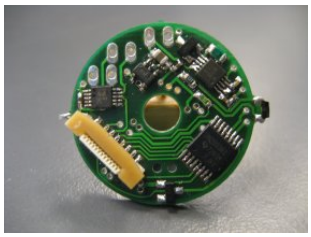
3.5.3 Specific Board

We made three implementations the Specific board. The first one is called Testing board and it embeds LEDs for visual debugging of the system. The second one is called Actuator board V1 and it controls a stepper motor. Finally, the third one is called Actuator board V2 and it controls a brushless DC motor. The Testing board and Actuator board V1 operate from a system voltage of 5V and the Actuator board V2 operates from a system voltage of 11.1V. Fig. 3.6 shows the three implementations of the Specific board.

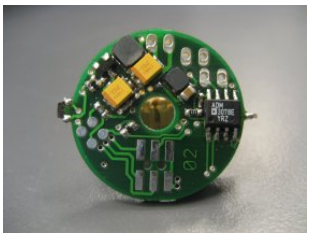


(a) Testing board (front).

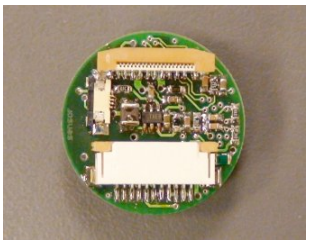
(b) Testing board (back).



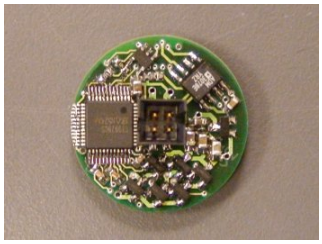
(c) Actuator board V1 (front).



(d) Actuator board V1 (back).



(e) Actuator board V2 (front).



(f) Actuator board V2 (back).

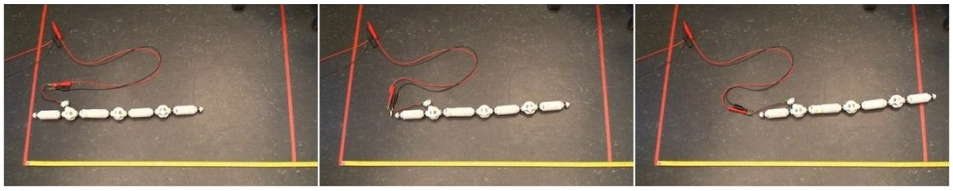
Figure 3.6: Implementations of the Specific board of the Odin robot. Fig. (a) and Fig. (b) show the front and back sides of the Testing board, which embeds LEDs for visual debugging; Fig. (c) and Fig. (d) show the front and back sides of the Actuator board V1, which controls a stepper motor; and Fig. (e) and Fig. (f) show the front and back sides of the Actuator board V2, which controls a brushless DC motor. The Specific board is a 25mm diameter printed circuit board (PCB).

3.5.4 Notes on the Implementation

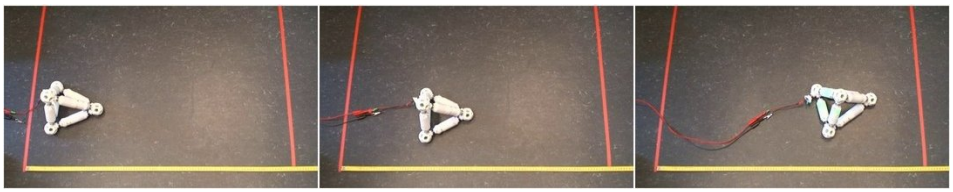
By combining a General board with an Actuator board V1 we have the required electronics for controlling a link which uses a stepper motor as actuator. In the same way, we may combine a General board with an Actuator board V2 for controlling a link which uses an brushless DC motor as actuator. The black links in Fig. 3.1 are an example of the former case.

Version 1 of the General and Specific boards were not easy to assemble. As result, in version 2, we embedded the power cables into many parallel lines sent together with the microcontroller signals (along with other improvements).

3.6 Locomotion Experiments



(a) Worm-like crawler locomotion.



(b) Tetrahedron locomotion.

Figure 3.7: Two different assemblies of Odin performing locomotion. Fig. (a) shows a worm-like crawler which travelled 1 meter from left to right in 2:02 min and Fig. (b) shows a tetrahedron structure which travelled the same distance from left to right in 3:56 min.

3.6.1 Settings

For these experiments, we configured Odin into a worm-like crawler (see Fig. 3.7a) and into a tetrahedron (see Fig. 3.7b). While the crawler is made of four linear actuator links and three joints, the tetrahedron is made of three linear actuators links, three structure links and four joints. The linear actuator links are equipped each with a General board V1 and an Actuator board V1, and the structure links are equiped each with a General board V1 and a Testing board.

In addition, we programmed both structures to perform the correspondent locomotion gaits. For the worm-like crawler we followed the algorithm proposed in [24], and for the tetrahedron we developed a custom algorithm which moves the center of mass of the structure so that displacement from left to right occurs. On these experiments we exclusively use local communication between links.

3.6.2 Results

The results of the previous settings are both structures performing the expected locomotion gaits. The worm-like crawler travelled 1 meter from left to right in 2:02 min, and the tetrahedron structure travelled the same distance from left to right in 3:56 min.

3.7 Discussion

As every link embeds one General board and one Specific board, and the General board is always the same, we are forced to re-use part of the components between designs. Moreover, if we are implementing two different actuator links (e.g., linear and rotational) which internally use the same motor, the electronics of the Specific board may also remain unchanged.

The last observation means that the design layout forces the homogeneity of the electronics to some extent, which is convenient when considering the high cost of heterogeneous systems made of a broad range of components. Nevertheless, the same advantage can be seen as a limitation, since new designs are constrained to the functionalities provided by the General board.

With the simple locomotion experiments shown in Section 3.6, we demonstrate that the electronic layout works properly.

3.8 Future Work

The current prototype of Odin comprises around 20 links in total, which are a combination of structure and linear actuator modules. Thus, our future work in this sense is to implement power links which will bring autonomy and mobility to the system. Overall, we want to have a larger system, with more types and instances of links. Therefore, in the near future, we want to produce 100 units of links (together with an appropriate amount of joints), which could be a combination of power, actuator and sensor modules.

Chapter 4

Communication

In this chapter we present a novel hybrid communication system for modular robots, based on inter-module buses that can connect on-demand to form arbitrary network topologies.

In addition to describing the implementation of this hybrid communication system, we analyse transfer rates and reliability, validating the results using a Spice [1] simulation and a proof-of-concept experiment performed on a hardware prototype.

Thus, we find the system is fast, since it has a potential to provide a maximum transfer rate of 9.9Mbps divided by the maximum bus length measured in meters, with buses as large as 256 modules. The system is also found to be small in size, power saving and reliable. These features, in combination with its flexibility, make hybrid communication suitable for modular robots.

4.1 Introduction

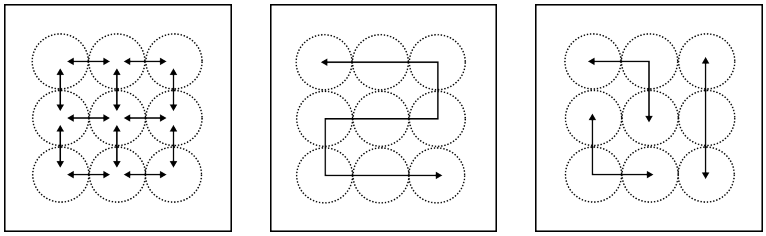
As mentioned in Chapter 1, modular robots are robots built from modules. The design of these systems is limited to homogeneous or heterogeneous modules, with basic functionality, which are able to combine into more complex entities. Thus, instead of designing a new robot for each task, a new suitable structure can be assembled from the available modules [1] [5].

Communication is central to modular robots. Local communication is used to figure out the topology of the robot and to coordinate tasks involving just local information [25]. On the other hand, global communication is needed for tasks which may require time critical coordination between distant modules of the system.

The hybrid communication approach provides a solution that is able to reconfigure the communication topology from small and local buses to long

[1] Spice is a general purpose circuit simulator that allows to check integrity of circuit designs and to predict circuit behaviour.

and more global buses, here called hybrid buses. The last is done on-demand, using the same buses and hardware, and maintaining the local transfer rates after the new topology is established. Figure 4.1 shows three examples of network topologies that hybrid communication can provide [26].



(a) Local Buses. (b) Global Bus. (c) Hybrid Buses.

Figure 4.1: Different kinds of communication buses within a modular robotic system. Here, modules are represented by circles and buses by arrows. In (a), every module communicates only with the adjacent neighbour modules using local buses. In (b), all the modules of the system communicate between each other through a common channel, a global bus. Finally, in (c), several non-global channels, hybrid buses, enable the communication between non-adjacent modules within the system. The global bus of (b), can be seen as an special case of hybrid bus.

Theoretical analysis indicates that the maximum transfer rates of the hybrid communication approach could be around 9.9Mbps divided by the maximum length of the hybrid buses measured in meters, with buses as large as 256 modules. Besides, the system is flexible, small in size, power efficient, and reliable.

This chapter introduces the hybrid communication approach, beginning with Section 4.3, which describes the requirements, design and implementation details of the communication system. Then, Section 4.4 describes the implementation of the hybrid communication approach in Odin and, Section 4.5 analyses the performance of the communication approach based on transfer rates: in general for any hybrid communication system and specifically for Odin. Afterwards, Section 4.6 validates the results of the performance analysis carried out on Odin, using a Spice simulation, and presents a proof-of-concept experiment performed on a hardware prototype of the same system. Finally, Sections 4.7 and 4.8 discuss potential issues that could appear when implementing the hybrid approach and the future direction of our work.

4.2 Related Work

The trade-off between local and global communication in modular robotic systems has previously been pointed out by several authors [26] [2]. Thus,

designers of modular robots have followed many approaches for taking advantage of both communication modalities.

A combination of two global communication mechanisms is used in the CEBOT self-reconfigurable robot [11]: serial wireless for self-reconfiguration (1Kbps) and parallel wired for coordination once connected (8 channels at 1Kbps). While it gives the CEBOT system two modalities of global communication, it does not allow modules to communicate locally.

Other designers have opted for implementing only local communication in their systems. CONRO [6] and ATRON [27] self-reconfigurable robots, have both infrared local communication at a transfer rate of 9600bps. For the ATRON, the main problem with this approach is the crosstalk between different modules.

Wireless communication is also used in the YAMOR [28] reconfigurable robot, where a complete bluetooth interface is fitted within large modules. Nevertheless, this kind of communication does not allow the modules to determine the physical location of their neighbours without using external sensors.

An alternative implementation of both communication modalities at once, was made in the M-TRAN II [8] self-reconfigurable robot. This system provides local (4800bps) and global (39Kbps) communication channels, by using three separated media and microcontrollers. In the M-TRAN II system, the dynamic topology of the global network (motivating the exclusion of termination impedance) is the main constraint for the transfer rates.

We believe the present work improves the communication capabilities of modular robotic systems, by allowing the implementation of both communication modalities at once: with high transfer rates, requiring only one type of communication hardware, using wired media, and demanding limited space.

4.3 The Hybrid Communication Approach

In a communication system for modular robots, a *local bus* provides a medium for exchanging messages between neighbor modules, whereas a *global bus* provides a medium for exchanging messages between all the modules in the system. Nevertheless, in the the context of the hybrid communication system, there is a third definition: a *hybrid bus* provides a medium for exchanging messages between non-neighbors or distant modules in the system (see Fig. 4.1).

4.3.1 Requirements

The main purpose for a hybrid communication system is to provide local, hybrid and global communication buses, all of them allowing high transfer rates. Further, the system must be small in size, scalable, power saving for

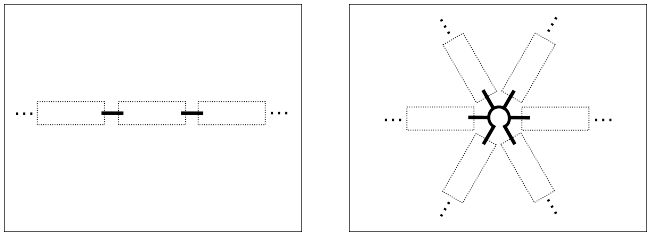
the sake of autonomy and reliable with regards to the noise ratio supported by the signals; which are the minimal set of features required by a modular robotic system.

4.3.2 Design

The idea behind the hybrid communication approach is to have fixed local buses. Then, on-demand, hybrid buses can be created by merging two or more local buses. If the hybrid bus is connecting all the modules in the system, it is a global bus.

Local Buses

The local buses are implemented as serial communication buses. Hence, modules must have a dedicated transceiver per local bus being accessed. Fig. 4.2 shows two examples of local buses, while Fig. 4.3 shows internal details of a module accessing two local buses.



(a) One-to-one local buses. (b) Many-to-many local bus.

Figure 4.2: Different types of local buses. Here, modules are represented by rectangles and buses by lines. In (a), the local buses connect two adjacent modules (neighbour-to neighbour communication). An example of this kind of system is homogeneous modular chain-type self-reconfigurable robots. In (b), a local bus connects many adjacent modules. In this kind of system, the local buses may be implemented outside the modules, within special types of modules acting as joints. An example of this kind of system is heterogeneous modular lattice-type self-reconfigurable robots.

Hybrid and Global Buses - Switching

For merging local buses, the modules must have an internal bus which can be attached to all the local buses being accessed. Thus, the internal bus is able to combine different channels into a common hybrid bus. The attachment can be achieved by using switching technology. Fig. 4.3 shows an example of module merging its local buses.

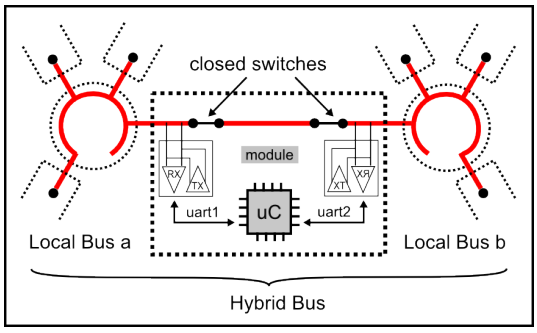


Figure 4.3: A module merging two local buses into a more global bus (hybrid bus) by closing its internal switches. If another adjacent module also closes its switches, the hybrid bus grows. When the hybrid bus is able to communicate all the modules within the system, it becomes a global bus. Here, the buses are represented by a single line, in spite of they are conformed by two wires.

Termination Impedance

The termination impedance is crucial for achieving high transfer rates, without decreasing the reliability of the communication system [29]. As the local buses are to be merged in a more global medium, the ideal pure resistive termination overloads the transceivers acting as drivers as soon as more than two local buses are merged [30]. Therefore, and conforming to the power saving requirement, the chosen termination impedance for the hybrid communication approach is an AC termination [31]. Fig 4.4 shows a local bus terminated with an AC termination.

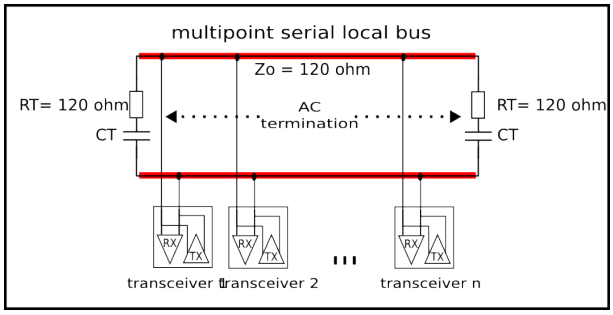


Figure 4.4: A local bus implemented using a transmission line with characteristic impedance $Z_0 = 120\Omega$. The AC termination consist of one resistor R_T in series with one capacitor C_T. R_T matches the characteristic impedance of the line, $Z_0 = 120\Omega$, and C_T is dependant on the expected maximum length of the hybrid buses. The AC termination is a balance between signal quality, power requirement, and transmission rates.

4.3.3 Implementation

Local Buses

For the sake of transmission rates, robustness, and transmission distances, the physical layer protocol used is RS485. As RS485 transceivers are characterized by unit loads, in the sense of electrical power consumed from the signal being transmitted, the transceivers included in the modules should have 1/8 of load, to create hybrid buses as large as 256 modules.

Hybrid and Global Buses - Switching

Considering the RS485 wiring guidelines [32], the cables used to implement local and hybrid buses (the internal bus within the modules), should be twisted pair and have a characteristic impedance of 120Ω. Moreover, for keeping low power consumption and high side switching capabilities, the buses should be extended with analog switches. Hence, the closed resistance should be as low as possible, then not to have important reflections nor voltage drops across the hybrid buses. Finally, the switches should be located at the RS485 side of the transceivers (not the UART side), as shown in Fig. 4.3.

Termination Impedance

When terminating a transmission line with an AC termination, two components are placed in series at the termination impedance: one resistor, R_T, and one capacitor, C_T (see Fig. 4.4).

R_T matches the characteristic impedance of the line (120Ω), and C_T is selected to be less or equal to the round trip delay of the cable divided by the characteristic impedance [31]. The round trip delay is defined as twice the time required for a signal to travel until the end of a line.

Thus, C_T is calculated as:

$$C_T = \frac{1}{120} \frac{2\, l_{cable\_max}}{0.66\, c} \tag{4.1}$$

where c represents the speed of light, $0.66\, c$ is the typical propagation velocity of a twisted pair cable, and l_{cable_max} is the maximum expected length of the hybrid buses.

Ideally, there should be just two AC terminations per local bus: one at each end. Therefore, in systems like Fig. 4.2a, the termination could be embedded within the modules; while in systems like Fig. 4.2b, the termination should be placed someway outside the modules.

4.4 Odin's Hybrid Approach Implementation

In the case of Odin, the local buses are implemented outside the links, within the joints. As recommended in Section 4.3, two AC terminations are placed per local bus, one at each end. Fig. 4.5 depicts the implementation of the local buses within the joints of Odin.

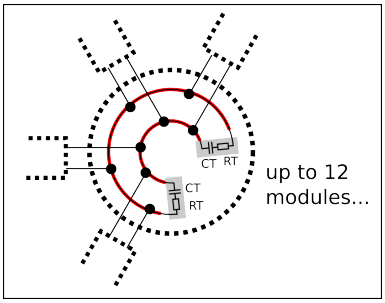


Figure 4.5: Local buses of Odin. One AC termination is placed at each end of the bus. The maximum number of links to be connected to the local bus is 12, and each link contributes 1/8 of load to the bus. The value of the capacitor C_T of the AC termination is determined by the maximum expected length of the hybrid buses (not the length of the local buses).

The maximum numbers of links to be connected to each local bus is 12. Further, two transceivers are implemented per link, one at each end. Finally, the stub of the transceivers is forwarded from the links to the joints through the connectors. Fig. 4.3 depicts the interior of an Odin's link connecting two joints.

4.4.1 CCP: The Lattice of Odin

A lattice is a mathematical concept used in crystallography, the science that describes the ways in which atoms and molecules are arranged in crystals [22]. As lattice-type modular robots are inspired by the discrete arrangement of atoms, the same concept is used to describe their configurations. Odin is a heterogeneous modular CCP lattice robot.

CCP lattice means that, if we imagine a 3D space grid, points are placed at each corner of a cube and also at every center of the six faces enclosing it. In crystals, these points are coincident with the location of atoms, but in Odin, they are coincident with the centers of joints. Moreover, the connectivity of atoms touching other atoms in crystals is represented by links in Odin. Fig. 4.6 depicts the CCP lattice arrangement of Odin.

In this section we perform a light analysis of the lattice of Odin, which will help us to estimate some parameters required for the next sections.

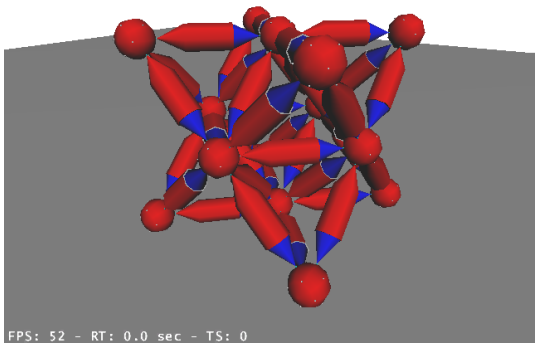


Figure 4.6: CCP lattice arrangement of Odin. If we continue expanding the size of the arrangement, by attaching more links to the joints, the average number of links per joints will converge to 12. The image was taken from the USSR simulator (Unified Simulator for Self-Reconfigurable Robots), which includes a model of the Odin system.

Number of joints versus size of the structure

Assuming a cubic expansion of the structure of Odin, when increasing the number of links and joints (e.g. Fig. 4.6); we can find a relation between a_n, the normalized edge length of a cube enclosing the structure, and the number of joints composing the system.

To begin, we define imaginary spheres concentric with the joints within the volume of the structure. These virtual spheres have a volume $V_b = \frac{4\pi}{3}\left(\frac{m}{2}\right)^3$, where m is the distance between the centers of two adjacent joints in the system. Thus, we calculate the number of joints, b_V, packed within the volume of the cube, $V = a^3$, as:

$$b_V = \frac{0.7405a^3}{V_b} = 0.7405\frac{6}{\pi}\left(\frac{a}{m}\right)^3 = 0.7405\,\frac{6}{\pi}\,{a_n}^3 \tag{4.2}$$

where the factor 0.7405 represents the spheres' packing efficiency in the CCP lattice [22].

Number of links versus size of the structure

CCP indicates the number of links per joint is 12 when the joints are located within the volume of the structure. Thus, the total number of links composing the system, N, can be approximated as:

$$N = b_V\frac{12}{2} = 0.7405\,\frac{6^2}{\pi}\,{a_n}^3 \tag{4.3}$$

Average number of links per joint versus size of the structure

A consequence of the cubic and quadratic nature of volume and surface, respectively, is that the ratio $r = \frac{volume}{surface}$ grows proportionally to the size of the structure [33]. Thus, we can approximate the average number of links per joint, n, as:

$$n = 12\frac{V_n}{S_n + V_n} = 12\frac{a_n}{6 + a_n} \tag{4.4}$$

where the term $V_n/(S_n+V_n)$ is the proportion of volume of the structure. Here we can see that when a_n approaches infinite (pure volume), n tends to 12 [33].

Hybrid buses length versus size of the structure

Assuming that each link merging local buses attaches $n - 1$ links to the hybrid bus (with the exception of the first one), the length of a hybrid bus communicating N links can be approximated as:

$$l = \frac{N}{n-1} - 1 \tag{4.5}$$

Note that l is not the physical length of the cable, but the number of links acting as bridges between local buses (internal switches closed).

Discussion about structural analysis

We have ignored the contribution of links and joints placed in the surface of the structure. Nevertheless, the previous equations are valuable estimations of n and l when the number of links, N, grows; which is specially useful in a system aiming for scalability.

4.4.2 Maximum hybrid bus length l_{max}

From section 4.3, we know the maximum number of links we can communicate with a single hybrid bus is 256. Therefore, by combining the equations (4.3), (4.4) and (4.5), we can get an estimation of the maximum length of the hybrid buses: $l_{max} \approx 80$ closed links. The last can be translated into a physical cable length of:

$$l_{cable\_max} = l_{max}0.2m = 16m \tag{4.6}$$

where $0.2m$ is the length of the bridge cable embedded on each link of Odin (worse case scenario).

Finally, the assumption of a maximum of 256 links connected to a hybrid bus must be handled with care. If two transceivers are implemented per link, 2/8 instead of 1/8 of load could be attached to the bus. Therefore, the

hardware should be able to disable one of the transceivers, when the link closes its switches or when it is communicating through the same hybrid bus on both ends.

4.4.3 Value of C_T for Odin

As we know the maximum expected length of the hybrid buses for Odin, l_{cable_max}, we can now calculate the value of C_T; the capacitor to be used in the AC termination of the local buses. Thus:

$$C_T = \frac{1}{120}\frac{2*16}{0.66\,c} \approx 1nF \tag{4.7}$$

4.5 Performance Analysis

The transfer rates affect the overall performance of a system. Instead of constraining the maximum length of the extended buses to be created by the hybrid communication approach, we are more interested in determining the maximum transfer rates we are able to communicate at, when having a defined structure.

4.5.1 Maximum Transfer Rate for A System In General

For digital signals, a rule of thumb of AC terminated transmission lines says that: the time constant $\tau = R_T\,C_T$ should be less than or equal to 10% of the bit width [31]. Therefore, the maximum transfer rate of the hybrid communication approach, tr_{max}, is determined by:

$$\begin{aligned} tr_{max} = \frac{1}{10}\frac{1}{120\,C_T} &= \frac{0.66\,c}{20\,l_{cable\_max}}\,bps \\ &= \frac{9.9}{l_{cable\_max}}\,Mbps \end{aligned} \tag{4.8}$$

4.5.2 Maximum Transfer Rate for Odin

Following the last section, we can now calculate the maximum transfer rate, tr_{max_odin}, to be achieved by the Odin modular robot:

$$tr_{max\_odin} = \frac{0.66\,c}{20*16} \approx 620\,Kbps \tag{4.9}$$

4.5.3 Consequences of Bad Topology

The hybrid buses created by the communication approach may present a topology similar to the bus at Fig. 4.7. There, we can see that one

transceiver is driving more than one local bus, which is discouraged by the RS485 wiring guidelines [32].

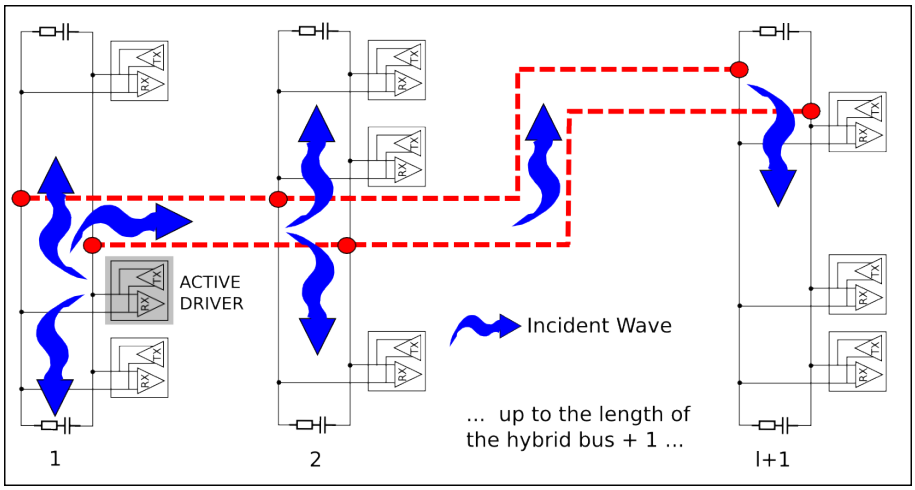


Figure 4.7: Example of topology created by hybrid buses. The transceiver acting as driver is coloured grey and the incident wave (not the reflexions) of the digital information is drawn with thick curly arrows. The active transceiver is driving more than one bus (AC terminated), and the local buses may be joined at other points than the extremes or the middle.

Nonetheless, those recommendations emphasize the load applied to the drivers, which is properly avoided by the AC terminations; and the reflexions generated in the connections points, which should be minimal if the switches extending the local buses have very small resistance.

4.6 Experiments

By having a concrete implementation of the hybrid communication approach in the Odin modular robot, we are able to perform simulations and experiments. The last allow us to validate the expected transfer rates and the capabilities of the communication system.

4.6.1 Spice Simulation

Settings

The simulations were performed in the Spice like simulator, LTSpice[2], by using trivial components. The only exceptions were the arbitrary behavioural

[2]http://www.linear.com

voltage source used for the signal generator, and the transmission lines elements used to simulate the cables' behaviour. Fig. 4.8 shows the block diagram of the simulation.

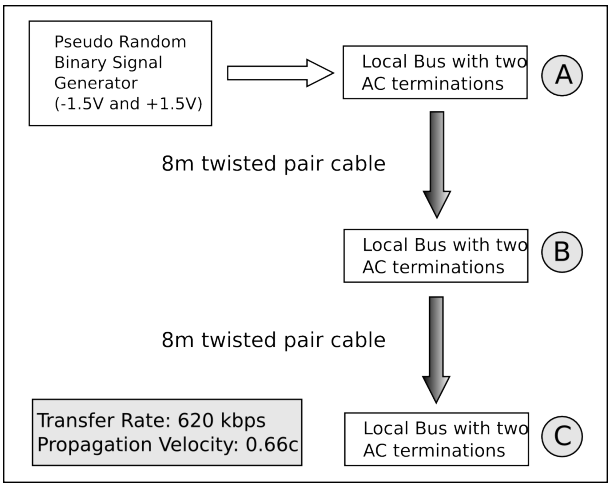


Figure 4.8: Block Diagram of the Spice Simulation. The Pseudo Random Binary Signal Generator (PRBS) information is coupled directly to the first local bus (A); then it is forwarded to the second local bus (B), which is connected by 8 meters of cable; and finally, it is forwarded to the third local bus (C), which is connected by 16 meters of cable (8+8 meters).

The Pseudo Random Binary Signal Generator (PRBS) emulates the behaviour of a RS485 transceiver. Thus, the signals generated are $\pm 1.5V$ differential, which is the minimal output that several commercial transceivers can deliver; and the transfer rate for sending bits was set to 620Kbps, which is the maximum transfer rate estimated for Odin. Further, the rise and fall time of the digital signals are set around 35ns, which determines a realistic slope generated by fast transceivers; and the source impedance is 10Ω for both differential outputs.

The transmission line elements modeling the cables match the characteristic impedance (120Ω) and the propagation velocity ($0.66\,c$) of twisted pair cables. The characteristic impedance can be set directly in the Spice component, while the propagation velocity is implicitly set by the trip delay parameter, *td*, which is the time required for a signal to travel until the end of a line. Thus, considering a length of 8 meters per cable segment (see Fig. 4.8), *td* was set to 40ns (8/0.66c).

Results

Reflexions As expected, the use of non-ideal AC terminations in the local buses produces reflexions in the signals. Fig. 4.9 shows the same digital

signal, at first ideally transmitted, and then sampled at one extreme of each local bus: A, B and C (see Fig. 4.8).

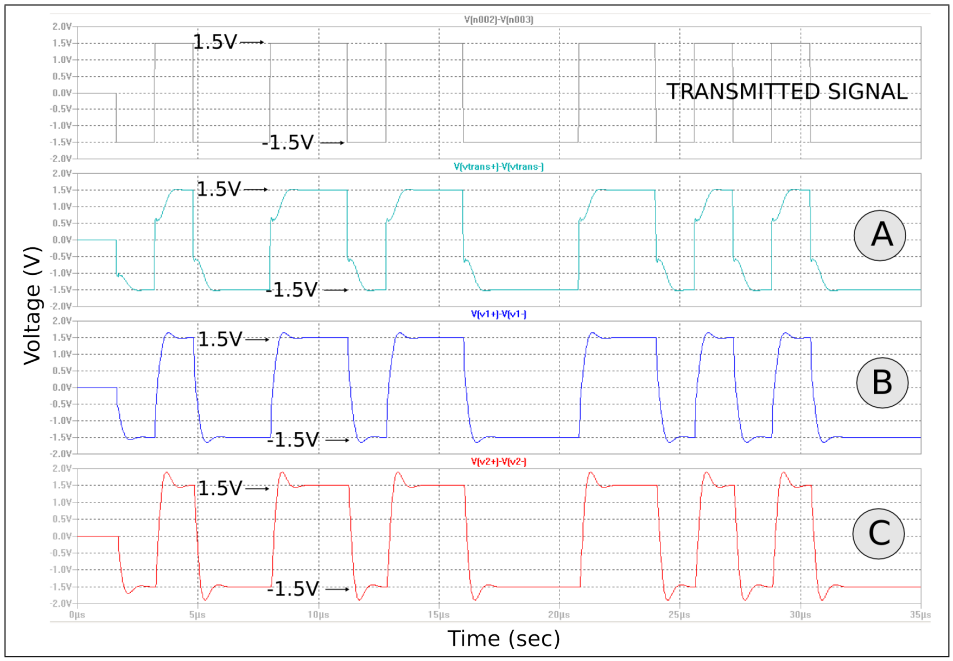


Figure 4.9: Digital signal ideally transmitted, and then sampled at one extreme of each local bus: A, B, and C. The non-ideal AC termination produces reflexions, which are successfully damped before the pulses end. The maximum trip delay is the time required for a signal to travel 16 meters in twisted pair cable. In the hybrid approach, the bit width is 20 times the maximum trip delay, and the time constant $\tau = R_T C_T$, is 2 times the maximum trip delay.

Thus, we see that all the reflexions are damped before the pulses end, which is a consequence of the guidelines followed for choosing the maximum transfer rate. With equation (4.8), we implicitly limit the *minimum* bit width to 20 times the maximum td (the time required to travel 16 meters).

Further, the time constant $\tau = R_T\, C_T$ is fast enough (2 times td) to produce overshoots and undershoots, more appreciable at the farthest end of the hybrid bus (C). The last is a typical pulse response of the AC terminations [34].

Current Requirements To continue, the AC termination requires less current than the pure resistive termination, as shown in Fig. 4.10. There, we see that the peak values of the current are around $\pm 50mA$, which is within the range of possibilities of commercial RS485 transceivers.

Thus, from Fig. 4.10 we conclude that even if the voltage being trans-

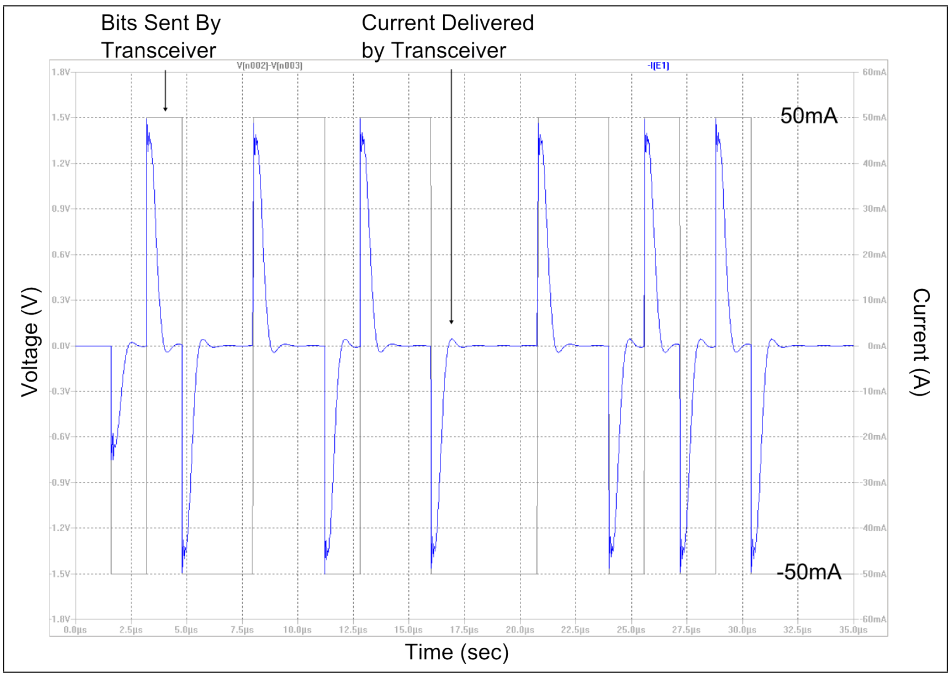


Figure 4.10: Current being delivered or received by one of the differential outputs of the modeled transceiver. The current is positive and negative, but is always provided by the same same transceiver (from alternated differential outputs). The peaks achieved are within the capabilities of commercial transceivers.

mitted is higher, meaning that the current required could be higher, the drop of voltage as consequence of overload would still be high enough not to produce transmission errors.

Reliability An Eye Pattern gives us an evaluation of a digital system performance at a glance [35]. Thus, Fig. 4.11 shows the Eye Pattern obtained from the Spice simulation. There, we sampled the same signals as Fig. 4.9, at the points A, B and C, but now against an appropriate synchronizing clock signal (not time in the horizontal axis). The clock signal is a ramp of voltage with amplitude 1V and frequency equal to 620kHz divided by 2 (ramps per second). The last frequency allows to capture two eyes in the Eye Pattern.

In Fig. 4.11, the distances nm indicate the amount of noise acceptable by the signals, before getting errors in the transmission (when the signals fall to $\pm 200mV$). These measurements decrease when increasing the length of the transmission line. On the other hand, the distances sr indicate the safety sampling interval, then not to fall into the slope segments where the signal can be wrongly interpreted. The intervals increase when increasing

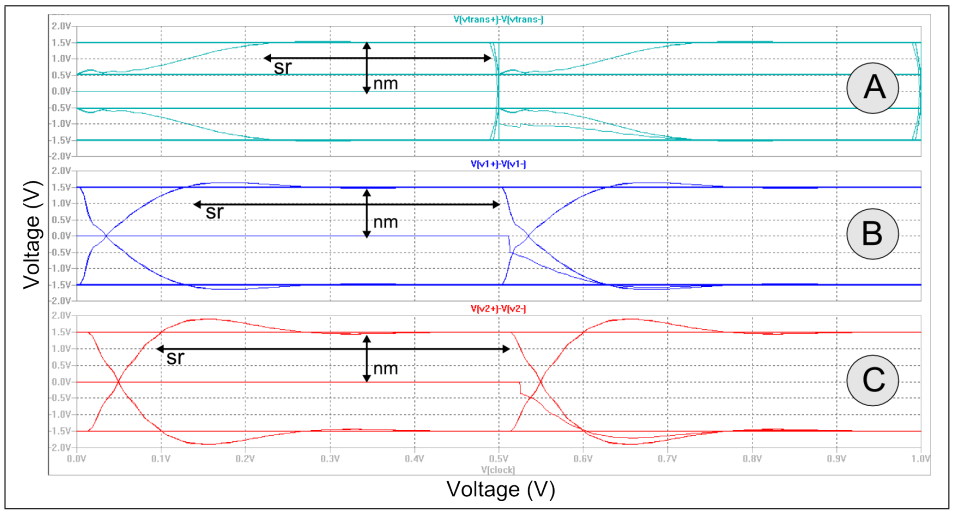


Figure 4.11: Eye Pattern of the Spice simulation. The digital signals are sampled at one extreme of each local bus: A, B and C, but now against an appropriate synchronizing clock signal (not time in the horizontal axes). The bigger the distances nm and sr (or the more open the eye), the better the quality of the signal.

the length of the transmission lines.

We have to recall that in these measurements we excluded the resistance of the cables, which has an impact on the distances nm. Therefore, even if the results indicate us that the greater the length of the cables, the better the quality of the signal (smaller ratio nm to sr), an attenuating effect will be added when using non-ideal lines.

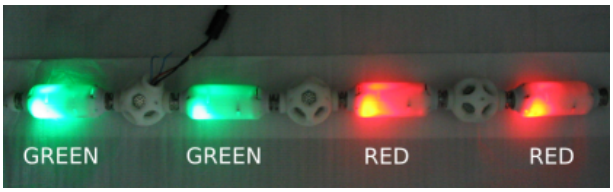
4.6.2 Proof-of-Concept

Settings

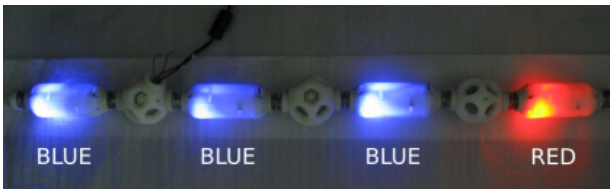
For the proof-of-concept we connected 4 Odin modules in a row, by using 3 joints that forward power and communication signals between modules. The modules are equipped with internal PWM-controlled RGB leds, which emulate the interface with future actuators of Odin. Fig. 4.12 shows the described arrangement.

Afterwards, we programmed the two middle modules to execute any of the following orders: light up with some colour, stop lighting up, open switches, or close switches. On the other hand, the two extreme modules are programmed to send those orders to their neighbours, and to execute the orders at the same time.

The leftmost module orders to blink green, and the rightmost module orders to blink red (see Fig. 4.12a). Here the blinking (light up and stop



(a) Two local buses: green and red.



(b) A hybrid bus: blue, and a local bus: red.

Figure 4.12: Local and Hybrid communication between modules of Odin. Fig. (a) and (b) are the same system photographed at different instants of time. In Fig. (a) two pairs of modules are performing local communication: green and red buses. In Fig. (b) the three leftmost modules are performing hybrid communication: blue bus; and the rightmost two modules, still have local communication between them: red bus (not completely shown here).

lighting up) is executed in coordination with the modules sending the orders. Finally, the leftmost module has a peculiarity: after sending some blink green orders to its neighbour, it sends one close switches and several blink blue orders (see Fig. 4.12b).

Results

The result of the previous setting is that the modules begin by establishing two local buses: green and red buses in Fig. 4.12a. Afterwards, when the leftmost module decides, the green local bus is extended to become a hybrid bus: blue bus in Fig. 4.12b; allowing communication between the 3 leftmost modules instantaneously.

Here, we have to recall that all the buses can be coordinated independently. We made them blink at the same time, for the sake of descriptive information from pictures. When making the buses blink asynchronously, we could see in the second scenario (Fig. 4.12b) that: even when the hybrid bus is established (blue bus), the local communication between the last two modules is not lost (red bus).

4.7 Discussion

Hybrid communication is specially useful when the system tasks can be split in roles [36] [37]. In that case, every role (e.g. leg, arm, or spine) can be coordinated by an specific bus, which has a highly optimized topology for the duty on hand. As the task or role changes, the communication topology can be once more optimized.

The main drawback of the hybrid communication approach is that, for small systems, communication between distant modules may not always be possible. A long hybrid bus connecting two extreme modules, could eventually divide the group into two blocks, therefore avoiding the creation of crossed hybrid buses. This problem may also appear in big systems presenting narrow passages.

In case the transfer rates provided by the hybrid communication approach are not enough for a specific system, the switching mechanism could be accompanied with bi-directional repeaters. Now, as the drivers load will be limited to one local bus, the local buses could be implemented with pure resistive terminations. Thus, the transfer rates may achieve the maximums estimated for the RS485 standard, which are 10Mbps or more considering the short distances covered by the local buses.

Nonetheless, even if the electrical isolation provided by repeaters will improve the response time of the overall system, the power required to drive every local bus will increase. Keeping the AC terminations will limit the transfer rates (higher than without repeaters) but will also keep the power saving features of the system. Thus, there is a trade-off between transfer rates and power saving.

Another way to increase the maximum transfer rates is to have a higher density of modules connected to the local buses. Odin is a good example of dense system, connecting up to 12 modules at once. Overall, these kinds of robots require less cable extension to communicate the upper limit of 256 modules.

To continue, a common aim of modular robotic systems is to reduce the size of the cells, and so the cable lengths needed for connecting higher number of modules. The last will translate into higher transfer rates when implementing hybrid communication.

Finally, as the communication technology evolves, we can see that transceivers decrease their load. Future improvements in chips will increase the maximum number of modules connected to the hybrid buses, making the hybrid communication approach a more scalable proposal.

4.8 Future Work

While reconfiguring the communication topology, the involved modules may not be allowed to perform coordinated tasks. Therefore, reconfiguration delay is an important parameter to have in mind when performing critical tasks, and is the future direction of our analyses.

Furthermore, the future work is also focused on testing the communication system in realistic applications, using Odin with higher number of modules involved, and where different kinds of communication topologies are required simultaneously.

Chapter 5

Communication Efficiency

In this chapter we compare analytically the efficiency of local and global communication within modular robots and discuss how hybrid schemes (providing extended local buses) could outperform the former communication modalities. For this analysis, we use the information transmission time from many to many modules as metric of communication efficiency.

Along this chapter, we develop a model for local communication within modular robots (i.e., the information diffusion process) and we use an available model of global communication for mobile robots which is also suitable for modular robots. In addition to analyzing efficiency, we apply our analysis to existing modular robots and validate our local communication model using the USSR[1] simulator.

NOT FINISHED... We conclude that local communication is convenient when two or more modules need to transmit information recurrently, global communication is convenient when one module needs to broadcast information to many modules at once and hybrid schemes generally outperform the former strategies when tasks can be executed by roles within the system.

5.1 Introduction

Traditionally, different communication strategies for modular robots are preferred for performing different tasks. For instance, local communication is used to obtain information about the topology of the structure, while global communication is considered a faster way to share time critical information between distant modules. Nevertheless, besides this common sense of convenience, no theoretical comparison has been done on communication efficiency for modular robots.

In this chapter we compare analytically the efficiency of local and global communication approaches by using the information transmission time from many n_f to many n_e modules within the system. As we already have a

[1] Unified Simulator for Self-Reconfigurable Robots.

suitable global communication model [26] for modular robots, we put most of our efforts at developing a local communication model which should be general enough to represent the many types and implementations of modular robots around the globe.

We begin this chapter with Section 5.2, which describes the analysis that motivated this work and its differences with the present analysis. Then, Section 5.3 shows the elaboration of a local communication model for modular robots, which is applied to existing systems for giving examples of usage. To continue, Section 5.4 presents the global communication model to be compared against the local communication model, and Section 5.5 performs such comparison. After that, Section 5.6 validates the local communication model by using the USSR simulator [38] with a model of the Odin robot and, finally, Sections 5.7 and 5.8 discuss the implications of this analysis and present our future work with regards to this work.

5.2 Related Work

This work adapts and complements the analysis made by Yoshida *et al.* in [26] for distributed mobile robotic systems. Concretely, we apply the same modelling methodology for communication and reuse most of the mathematics formulated in [39, 40, 41, 26, 42]. Therefore, we take care of the new simplicities and complexities brought by modular robots to an analysis already made for similar systems.

Although mobile robotic systems are in many senses similar to modular robots, there are differences that make it worth a new analysis. The important differences between modules of modular robots and robot units of mobile robotic systems are:

- modules does not move relatively to its neighbours (we do not consider the self-reconfiguration process),
- modules are not freely distributed in space, but follow a well defined pattern (e.g., chain or lattice), and
- the spatial density of modules is usually higher than the spatial density of mobile units.

Consequently, parameters like displacement velocity and output communication range mentioned in [26] are not relevant in this study.

5.3 Analysis on Local Communication

5.3.1 Local Communication Model

Our local communication model for modular robots is based on the average number of neighbours connected to one interface of one module. Therefore,

the parameters to consider in this model are:

- i: number of communication interfaces of one module,
- m: maximum number of neighbours connected to one interface of one module,
- n: average number of neighbours connected to one interface of one module,
- p: probability of having one neighbour connected to one interface of one module, $p = \frac{in}{im} = \frac{n}{m}$,
- p_e: probability of information output from a module,
- c: maximum number of neighbours one interface can simultaneously receive information from,
- n_t: total number of modules in the system,
- n_e: number of modules the information is transmitted to,
- $r(t)$: ratio of informed modules at time t.

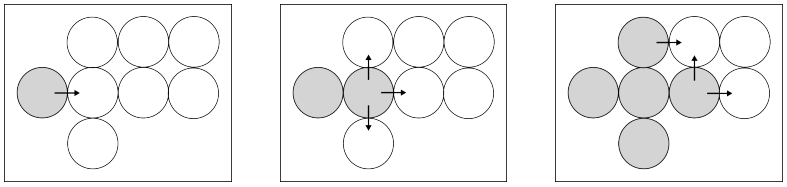
Notice that the parameters i, m and c are purely defined by the mechanical design of the modular robot. In addition, even though this model assumes that connectivity of modules is homogeneous (i.e., the same amount of interfaces and communication capabilities), the functionalities of the modules may be heterogeneous (as in the case of the Odin robot).

Next, we define $Imods$ as the modules that have received the information of interest and $Nmods$ as the modules that have not received such information yet. Thus, the ratio of $Imods$ at time t is represented by $r(t)$. Said that, we assume that the information diffusion process occurs as follows:

1. $Imods$ modules send information across their interfaces which reaches directly connected neighbours,
2. if these neighbours are $Nmods$ modules, they become $Imods$ modules,
3. step 1 is repeated now by a larger group of $Imods$, until $Imods = n_e$.

In this model we assume that one time unit is long enough to process any information available to one module and that collisions happen if more than one message is received across the same communication interface (i.e., $c = 1$) during that period of time. Consequently, the information transmission time, T_{loc} is defined as the number of time units before the information is received by n_e modules.

As many processes like the one described above may happen simultaneously at different parts of the system, the number of transmitting modules n_f is not relevant for this model, but rather the number of receiving modules



(a) First Stage. (b) Second Stage. (c) Third Stage.

Figure 5.1: Model of local communication. In this model, informed modules spread information across their communication interfaces, which reaches non-informed modules. After that, the process is repeated recurrently until the information is received by a specific number of modules.

n_e [39, 26]. Fig. 5.1 shows a three-stages sequence of this communication model.

If modules are connecting one neighbour per communication interface there is not problem with interfering communication between modules (collisions). Nevertheless, there are systems where each interface electrically connect with many neighbours (e.g., the Odin robot).

5.3.2 Analysis of Transmission Time

For estimating the information transmission time we are going to get, at first, the probability that one non-informed module successfully receives information from at least one informed module. We call this probability: information transmission probability, P_{tx}.

By having P_{tx}, we can then estimate the rate $\Delta r(t)/\Delta t$ of newly generated $Imods$, which allow us to formulate the information diffusion process. For calculating P_{tx} we repeatedly use binomial and conditional probabilities.

Information Transmission Probability

Provided that modules are randomly connected in some configuration, the probability of having x neighbours connected to *one* interface of one module is given by the binomial probability:

$$P(x, p, m) = {}_mC_x \, p^x \, (1-p)^{m-x} \qquad \text{with } x = 0..m \tag{5.1}$$

Then, we are able to calculate the probability that y out of x modules already connected are sending information out, as:

$$P(y, x, p_e, p, m) = {}_xC_y \, {p_e}^y \, (1-p_e)^{x-y} \, P(x, p, m) \qquad \text{with } y = 0..x \tag{5.2}$$

And, finally, we are able to calculate the probability that at least one out of y modules sending information out is an $Imods$, as:

$$P(t, y, x, p_e, p, m) = (1 - [1 - r(t)]^y)\, P(y, x, p_e, p, m) \tag{5.3}$$

where we consider $r(t)$ as the probability that a single connected module is an $Imods$.

After that, and assuming interfering communication [40] or modules blocking each other when attempting more than c multiple transmissions through a single interface, the information transmission probability is defined by:

$$\begin{aligned} P_{tx}^{single}(t, y, x, p_e, p, m, c) = & \sum_{x=1}^{c} \sum_{y=1}^{x} P(t, y, x, p_e, p, m) \\ & + \sum_{x=c+1}^{m} \sum_{y=1}^{c} P(t, y, x, p_e, m, p) \end{aligned} \tag{5.4}$$

In other words, the information transmission probability through a single interface is equal to the sum of the probability of having $x \leq c$ neighbour modules connected to the interface and $y \leq x$ of those neighbours transmitting information, plus the probability of having $x > c$ modules connected to the interface but just $y \leq c$ modules transmitting information.

Simplification of P_{tx}^{single}: As mentioned in Section 5.3.1, our analysis assumes collisions when more than one message is received through a single interface. That constraint translates into $c = 1$ in our formulation, and so the information transmission probability simplifies to:

$$\begin{aligned} P_{tx}^{single}(t, y = 1, x, p_e, p, m, c = 1) = & \sum_{x=1}^{1} P(t, y = 1, x, p_e, p, m) \\ + \sum_{x=2}^{m} P(t, y = 1, x, p_e, p, m) = & \sum_{x=1}^{m} P(t, y = 1, x, p_e, p, m) \end{aligned} \tag{5.5}$$

Finally, the equation 5.5 can be further simplified, by following the mathematical development shown in Appendix A Section A.1, resulting in:

$$P_{tx}^{single}(t, p_e, p, m) = r(t)\, p_e\, p\, (1 - p_e p)^{m-1}\, m \tag{5.6}$$

Final formula for P_{tx}: By having P_{tx}^{single}, we can now consider the total number of interfaces of a module, i, since every interface can successfully receive information. Consequently, we calculate the probability that at least one out of i interfaces is successfully receiving information, as:

$$P_{tx}(t, p_e, p, m, i) = 1 - [1 - r(t)\, p_e\, p\, (1 - p_e p)^{m-1}\, m]^i \tag{5.7}$$

which is the information transmission probability we originally sought.

Transmission Time

We now calculate the increment of $r(t)$ per unit of time Δt, $\Delta r(t)$, as it corresponds to the percentage of newly generated $Imods$ at time t. As the increment $\Delta r(t)$ is in proportion to the ratio of $Nmods$, $1 - r(t)$ and P_{tx}, the information diffusion process is modeled by the following differential equation:

$$\frac{\Delta r(t)}{\Delta t} = P_{tx}(t, p_e, p, m, i)(1 - r(t)) \tag{5.8}$$

Nevertheless, since Eqn. 5.8 does not have a general solution, we simplify P_{tx} by using linear interpolation over the variable $r(t)$, between the points 0 and 1, what results in:

$$\frac{\Delta r(t)}{\Delta t} = (1 - [1 - p_e\, p\, (1 - p_e p)^{m-1}\, m]^i)\, r(t)\, (1 - r(t)) \tag{5.9}$$

whose solution is the logistic equation:

$$r(t) = \left(1 + \frac{1 - r(0)}{r(0)}\, e^{-(1-[1-p_e p(1-p_e p)^{m-1} m]^i)t}\right)^{-1} \tag{5.10}$$

and where $r(0)$ is the ratio of $Imods$ at time 0, $1/n_t$.

Thus, the information transmission time is given by:

$$T_{loc}(n_e) = \frac{-1}{1 - [1 - p_e p(1 - p_e p)^{m-1} m]^i}\, ln\left[\frac{n_t - n_e}{(n_t - 1)n_e}\right] \tag{5.11}$$

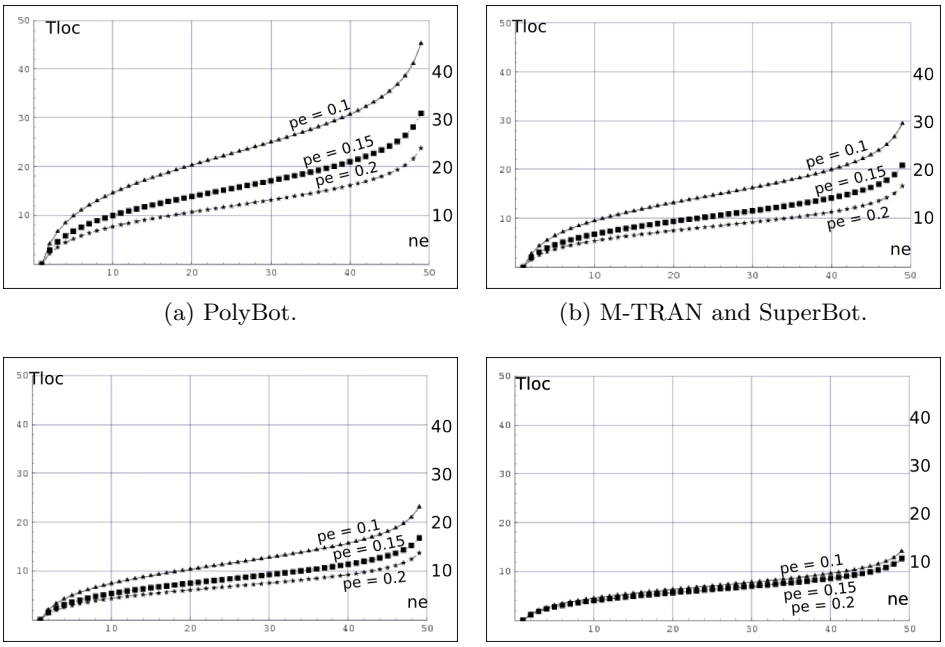
5.3.3 Applying the Model

Fig. 5.2 shows the curves of the information transmission time obtained for the models of five modular robots: PolyBot, M-TRAN, SuperBot, ATRON and Odin.

In Fig. 5.2 we see that the information transmission times for local communication, $Tloc$, increases monotonously with n_e. In addition, we can see that chain-type robots (here represented by PolyBot) are slower at propagating information by using local communication than lattice-type modular robots.

From Fig. 5.2 we can also see that information diffusion by using local communication is faster when the probability of information output from a module, p_e, is high. p_e is defined by the ongoing task and it helps the propagation of information by allowing informed modules to communicate more frequently.

Nevertheless, we should keep low values for p_e when using the local communication model, as it relies on a linear interpolation that translates into bad approximations when the values of p_e are high (see Appendix A Section A.2).



(a) PolyBot. (b) M-TRAN and SuperBot.

(c) ATRON. (d) Odin.

Figure 5.2: Information transmission times of five modular robots: PolyBot, M-TRAN, SuperBot, ATRON and Odin. PolyBot is modelled as having two interfaces $i = 2$ and at most one neighbour $m = 1$, M-TRAN and SuperBot are modelled with the same number of interfaces $m = 6$ and at most one neighbour $m = 1$, ATRON is modelled as having eight interfaces $i = 8$ and at most one neighbour $m = 1$ and Odin is modelled as having two interfaces $i = 2$ and at most 11 neighbours $m = 11$.

5.4 Analysis on Global Communication

5.4.1 Global Communication Model

We use the global communication model developed for distributed mobile robots in [26], since the characteristics of global communication are the same in modular robots. The only difference is that the global communication medium does not need to be wireless in the case of modular robots.

In such model, the authors assume that global communication is based on time-division multiple access (TDMA) to a common medium, so that the global communication process occurs as follows:

1. one module send information through a common medium, when a time slot is assigned either by a central manager or a token passing method,

2. the information reaches every module in the system,

3. step 1 is repeated until n_f modules are each assigned a time slot and, therefore, until all n_f modules have sent information.

In this model we assume that a time slot is long enough to send all the information available from one module and that a time slot equals a time unit defined for the local communication model, so the information transmission time counted in time slots can be directly compared with T_{loc}. Thus, the information transmission time for global communication, T_{glo}, is defined as the number of time slots before the information is transmitted by n_f modules.

As every time one modules access the common communication medium, the information reaches all the other modules in the system, the number of receiving modules n_e is not relevant for this model, but rather the number of transmitting modules n_f [26].

5.4.2 Analysis of Transmission Time

Based on hyper-geometric and conditional probabilities together with the expected value of the information transmission time for global communication, Yoshida *et al.* concluded that the information transmission time, T_{glo} is defined by the equation:

$$T_{glo}(n_f) = \sum_{i=n_f}^{n_t} \frac{{}_{i-1}C_{n_f-1}}{{}_{n_t}C_{n_f}}\, i \tag{5.12}$$

where n_t is the total number of modules in the system. Eqn. 5.12 can further visualized in Fig. 5.3, which shows the information transmission time, T_{glo}, for a system made of 50 modules.

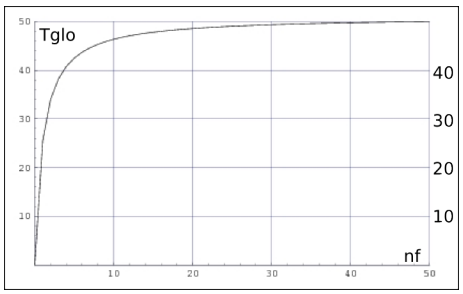


Figure 5.3: Information transmission time, T_{glo}, for global communication when $n_t = 50$. T_{glo} approaches n_t as soon as more than two modules, n_f, attempt to access the global communication medium.

Notice that the curve in Fig. 5.3 represents the global communication process at any modular robot, independent of the mechanical design.

5.5 Comparison of Transmission Times

As we mentioned in 5.1, we are going to compare the information transmission times T_{loc} and T_{glo} when the information is sent from n_f to n_e modules. In addition, as T_{loc} is independent of n_f and T_{glo} is independent of n_e, we are going to plot T_{loc} as a function of n_e and then we are going to overlap horizontal lines representing T_{glo} (which is constant with regards to the number of receiving modules).

5.5.1 Parameters of Evaluation

For this comparison we want to evaluate local communication under worse scenarios. To do that we use chain-type robots and also lattice-type robots with few neighbours connected to the interfaces. The last characteristic is represented by a low probability p in the model of local communication.

Said that, we compare local and global communication in systems made of 50 modules, which are: the PolyBot robot with $i = 2$, $m = 1$ and $p = 0.9$; the M-TRAN and SuperBot robots with $i = 6$, $m = 1$ and two values for p, 0.2 and 0.3; the ATRON robot with $i = 8$, $m = 1$ and two values for p, 0.2 and 0.3; and the Odin robot with $i = 2$, $m = 11$ and two values for p, 0.2 and 0.3.

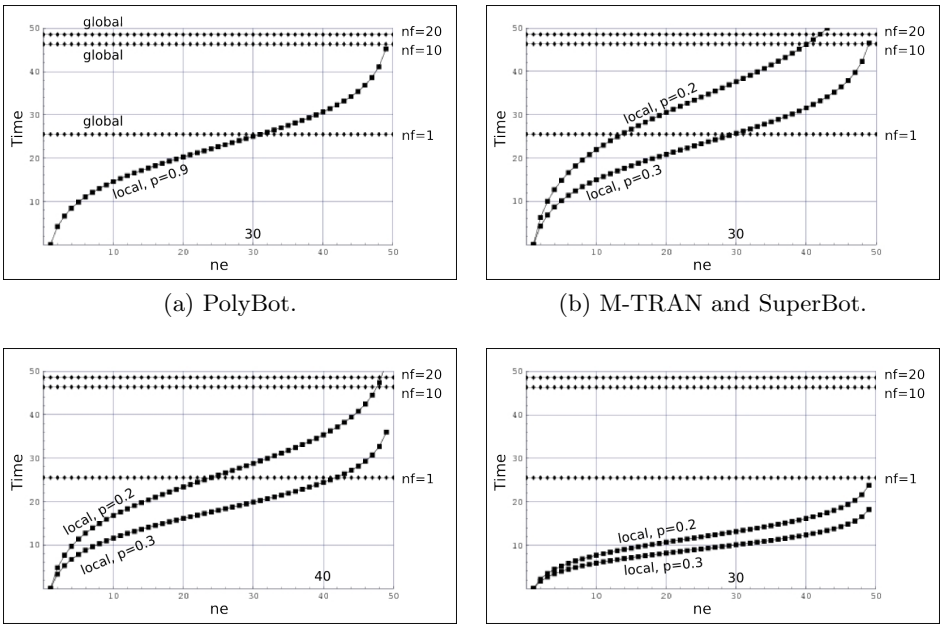
5.5.2 Results

By implementing the parameters mentioned above, we obtained the plots depicted in Fig. 5.4, which provide all the information needed to conclude about efficiency.

Assuming that one and just one module needs to send information to n_e modules in the system, we conclude from Fig. 5.4 that local communication is convenient: in the PolyBot robot when the number of receiving modules, n_e, is less than 30; in M-TRAN and SuperBot robots when n_e is less than 15 for $p = 0.2$ or less than 30 for $p = 0.3$; and in the ATRON robot when n_e is less than 20 for $p = 0.2$ or less than 40 for $p = 0.3$. On the other hand, local communication is always convenient for the Odin robot when having such amounts of neighbours connected to the interfaces (i.e., for $p = 0.2$ or $p = 0.3$).

To continue, if two or more modules are thought to transmit information, global communication is not able to outperform local communication anymore, due to information transmission times that quickly approach the total number of modules in the system (50 in this case).

Thus, we conclude that: on the one hand, global communication is convenient for implementing centralized-type of control approaches, which require only one module sending orders to its pairs and, on the other hand, local communication is convenient for implementing distributed control ap-



(a) PolyBot. (b) M-TRAN and SuperBot.

(c) ATRON. (d) Odin.

Figure 5.4: Information transmission times of five modular robots: PolyBot, M-TRAN, SuperBot,

proaches, where many modules need to continuously exchange information between each other.

5.5.3 Hybrid Schemes

As we visualized in Chapter 4, hybrid communication approaches providing extending local buses are convenient when splitting the system into sections performing common roles. For instance, consider a four legged walker. Assuming that the legs are made of many modules and the interconnection between the legs by few modules, it is convenient to implement global buses in the legs and local communication between the central controllers of each leg.

This conclusion, drawn from our comparison between local and global communication, is further supported by our observations in Section 5.3.3, where we concluded that local communication presents better transmission times in lattice-type than chain-type robots. The legs of our walker could be seen as chain-type assemblies.

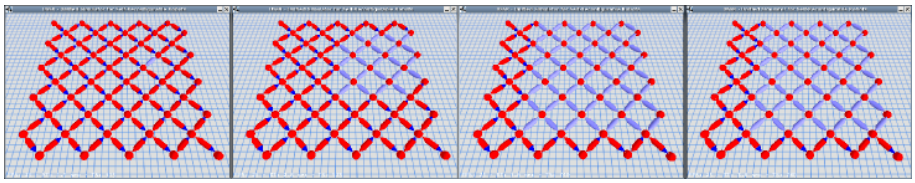
Ironically, based on 5.4, we should conclude that the system less beneficed by hybrid communication schemes is the Odin robot (local communication is always better); which is true as long as the assemblies are kept with high spatial density of modules. Nevertheless, chain-type assemblies are often

built by our system (as we have done in the experimental sections of previous chapters) and we do not want force the assemblies to use the complete connectivity of Odin all the time. We just want to have such possibility.

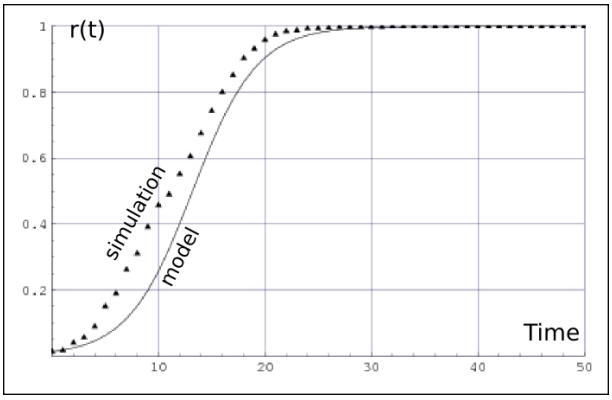
5.6 Simulation

For validating our local communication model we used the USSR simulator [38] with an embedded model of Odin. In this simulations we model the information diffusion process as informed modules trying to propagate information in a system where the non-informed modules are sending information out as mean of interference (for creating collisions). Non-informed modules send out information with probability p_e.

Next, Fig. 5.5 shows screenshots of a planar assembly built for the first simulation, which has parameters: $i = 2$, $m = 11$, $p = 0.204$ and $p_e = 0.1$.



(a) Information diffusion in a planar assembly of Odin.

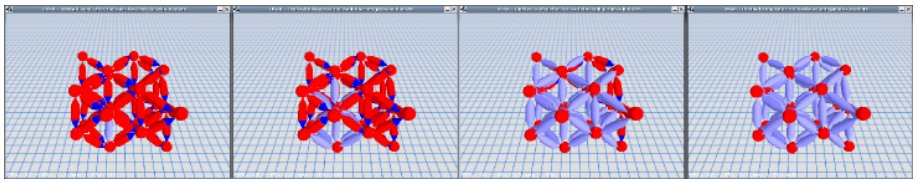


(b) Comparison between simulation and model results.

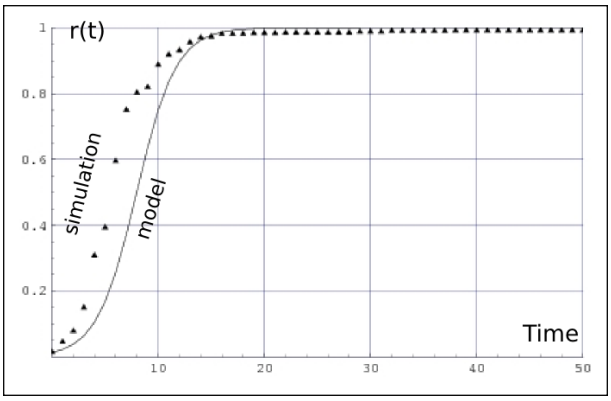
Figure 5.5: Simulation of the information transmission time, T_{loc}, performed on a planar assembly of Odin.

On the other hand, Fig. 5.6 shows screenshots of a cubic assembly built for the second simulation, which has parameters: $i = 2$, $m = 11$, $p = 0.477$ and $p_e = 0.1$.

In addition, Figs. 5.5 and 5.6 also show two plots matching the information transmission times, T_{loc}, obtained from simulations and from the local communication model. Thus, we confirm that our model approximates the



(a) Information diffusion in a cubic assembly of Odin.



(b) Comparison between simulation and model results.

Figure 5.6: Simulation of the information transmission time, T_{loc}, performed on a cubic assembly of Odin.

behaviour of information diffusion by local communication.

5.7 Discussion

We are conscious that using the Poisson distribution for modelling the spatial distribution of modular robots would end up with easier calculations. Nevertheless, we consider that the probability p is not small enough for accurate approximations [43]. In other words, we consider that the random connectivity of modular robots is not a Poisson process.

From Fig. 5.4 we realize that assemblies of dense structures, improves the information transmission times when using local communication. It means that chain-type robots are less efficient than lattice-type robots when using local communication, what matches our observations in Section 5.3.3.

Even though our local communication model does not works well for high values of p_e, it allows us to perform comparisons based on worse case scenarios, which is very useful for taking decisions about network topologies.

5.8 Future Work

In the near future we would like to perform simulations over a wider range of modular robots, so that more reliability is given to our model. In addition, we would also like to compare our model with information transmission times obtained from real experiments carried on the Odin system assembled in large structures. The last is going to be possible in the near future, since we are now building a set of modules that will enable us to assembly structures as large as 100 modules.

Chapter 6

Conclusion

This thesis described the electronics and communication system of Odin and performed an analysis evaluating the efficiency of local and global communication in modular robots. The electronics was developed to realize the concept of hierarchical robots by building Odin, the communication system was developed to provide flexible network topologies to such robot, and the analysis on efficiency was elaborated for giving theoretical support to the potential advantages visualized for the hybrid communication approach.

For the electronics we developed a layout which split the whole amount of components of the links into two boards: General and Specific. In the General board we embedded the components which are common to every link in the system (e.g., microcontroller) and in the Specific board we embedded the specialized hardware (e.g., battery monitor or motor driver). In this way we forced to some extent the homogeneity of the electronics, even though we are implementing a heterogeneous system. This approach for implementing electronics in heterogeneous systems could be further adopted by other robots.

For the communication system, we developed a hybrid communication approach which allows the system to change the communication topology on-line and which is appropriate for modular robots whose control algorithms are based on roles. We think the hybrid communication approach is suitable for modular robots, as it improves the transfer rates, flexibility, space requirements, power efficiency and reliability of the systems. The hybrid communication presented a concept for communication systems not explored previously. Besides, the hybrid communication approach can also be adopted by other modular robots, and the consideration for that labour are given in this work.

Even though the analysis on the communication efficiency was thought to give some theoretical support to the advantages visualized for the communication system at earlier chapters, we developed a local communication model which is flexible enough to be applied on a broad range of modu-

lar robots. This local communication model can be used together with an existing model for global communication, for taking decision about communication modalities in systems following specific mechanical designs.

Appendices

Appendix A

Mathematical Development

In this appendix we show the mathematical development of sections included in the main body of the thesis, which are of interest for readers looking for more detailed elaborations.

A.1 Binomial Theorem

The idea of this development is to simplify the equation 5.5 by shaping it into the binomial theorem's form. Said that, the equation,

$$P_{tx}^{single}(t, y = 1, x, p_e, p, m, c = 1) \sum_{x=1}^{m} P(t, y = 1, x, p_e, p, m) \tag{A.1}$$

expands to,

$$P_{tx}^{single} = \sum_{x=1}^{m} {}_mC_x \, p^x \, (1-p)^{m-x} \, {}_xC_1 \, p_e \, (1-p_e)^{x-1} \, r(t) \tag{A.2}$$

which can be arranged as,

$$\begin{aligned} P_{tx}^{single} &= \frac{r(t)\, p_e \, (1-p)^m}{1-p_e} \sum_{x=1}^{m} \left(\frac{p\,(1-p_e)}{1-p} \right)^x {}_mC_x \, {}_xC_1 \\ &= \frac{r(t)\, p_e \, (1-p)^m \, m!}{1-p_e} \sum_{x=1}^{m} \left(\frac{p\,(1-p_e)}{1-p} \right)^x \frac{1}{(m-x)!} \frac{1}{(x-1)!} \\ &= a \sum_{x=1}^{m} b^x \frac{1}{(m-x)!} \frac{1}{(x-1)!} \end{aligned} \tag{A.3}$$

where $a = \frac{r(t)\, p_e \, (1-p)^m \, m!}{1-p_e}$ and $b = \frac{p\,(1-p_e)}{1-p}$. Next, we can change variables from x to $z = x - 1$, expressing Eqn. A.3 as,

$$
\begin{aligned}
P_{tx}^{single} &= a \sum_{z=0}^{m-1} b^{(z+1)} \frac{1}{([m-1]-z)!} \frac{1}{z!} \\
&= ab \sum_{z=0}^{m-1} b^{z} \frac{1}{([m-1]-z)!} \frac{1}{z!} \\
&= \frac{ab}{(m-1)!} \sum_{z=0}^{m-1} b^{z} \frac{(m-1)!}{([m-1]-z)!\, z!} \\
&= \frac{ab}{(m-1)!} \sum_{z=0}^{m-1} {}_{m-1}C_z \, (1)^{[m-1]-z} \, b^{z} \qquad \text{(A.4)}
\end{aligned}
$$

which has the form of the binomial theorem over the variables 1 and b. Thus, Eqn. 5.5 can be finally simplified to,

$$
\begin{aligned}
P_{tx}^{single} &= \frac{ab}{(m-1)!} \sum_{z=0}^{m-1} {}_{m-1}C_z \, (1)^{[m-1]-z} \, b^{z} \\
&= \frac{ab}{(m-1)!} (1+b)^{m-1} \\
&= \frac{r(t)\, p_e\, p\, (1-p)^{m-1}\, m!}{(m-1)!} \left(1 + \frac{p(1-p_e)}{1-p}\right)^{m-1} \\
&= \frac{r(t)\, p_e\, p\, (1-p_e p)^{m-1}\, m!}{(m-1)!} \\
&= r(t)\, p_e\, p\, (1-p_e p)^{m-1}\, m \qquad \text{(A.5)}
\end{aligned}
$$

A.2 Consequences of Linear Interpolation

When developing our local communication model, we approximated the equation,

$$
P_{tx}(t, p_e, p, m, i) = 1 - [1 - r(t)\, p_e\, p\, (1-p_e p)^{m-1}\, m]^{i}\,, \qquad \text{(A.6)}
$$

with the equation,

$$
P_{tx}(t, p_e, p, m, i) = (1 - [1 - p_e\, p\, (1-p_e p)^{m-1}\, m]^{i})\, r(t)\,, \qquad \text{(A.7)}
$$

which is a linear interpolation of Eqn. A.6 over the variable $r(t)$ and between the points 0 and 1. The last means that our interpolation is accurate, as

long as Eqn. A.6 is approximately a line. The next Fig. shows a comparison between the linear approximation given by Eqn. A.6 and the real Eqn. A.7, varying $r(t)$ between 0 and 1 and considering different values of i.

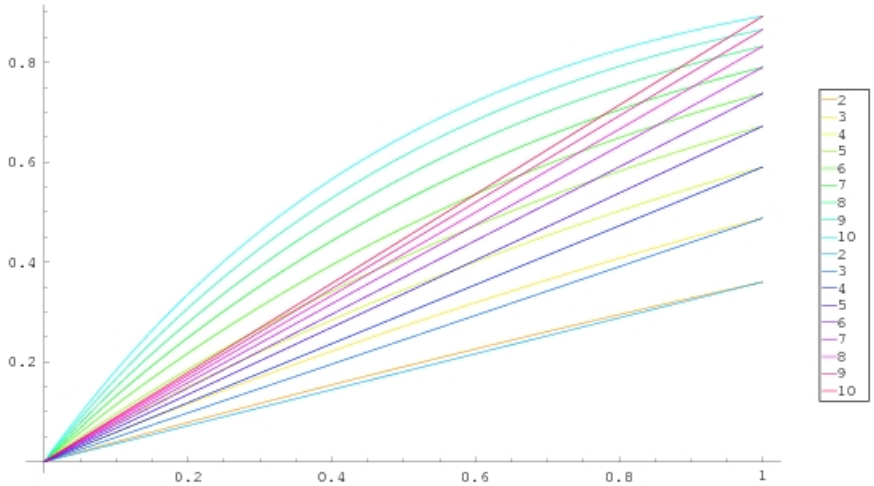
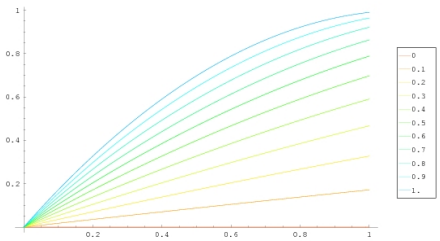


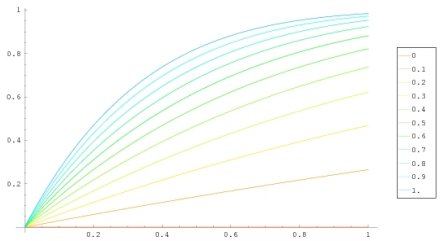
Figure A.1: Comparison between Eqn. A.6 (non-linear function) and Eqn. A.7 (linear function) for different values of i. The different values of i according to the different colors of the curves are shown at the right side leyend.

In Fig. A.1 we can see that the non-linearity of Eqn. A.6 is augmented with an increment of the values of i, which ends up with non-accurate approximations. Nevertheless, we have not found modular robots with more than eight interfaces in their modules.

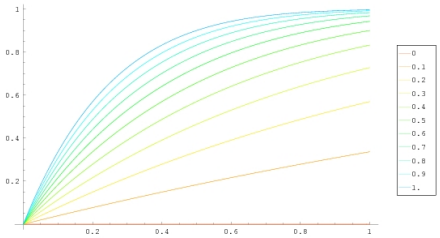
In addition, increasing the values of p_e also increases the non-linearity of Eqn. A.6, regardless we are modelling modules with few interfaces. This fact is shown in Fig. A.2, which depicts Eqn. A.6 for different values of p_e. Thus, we conclude that we should keep low values of p_e when using our local communication model.



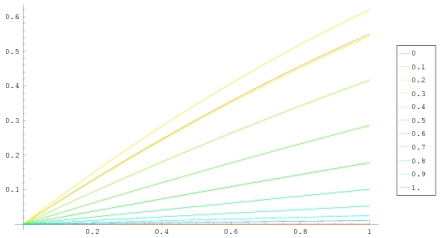
(a) Polybot ($i = 2$ and $m = 1$).



(b) M-TRAN and SuperBot ($i = 6$ and $m = 1$).



(c) ATRON ($i = 8$ and $m = 1$).



(d) Odin ($i = 2$ and $m = 11$).

Figure A.2: Non linearity generated on Eqn. A.6 by growing values of p_e, for modular robots with different values for i and m. The different values of p_e according to the different colors of the curves are shown at the right side leyend of each graph.

Appendix B

Schematics

This appendix presents the schematic diagrams of most of the electronics developed for Odin along the project period.

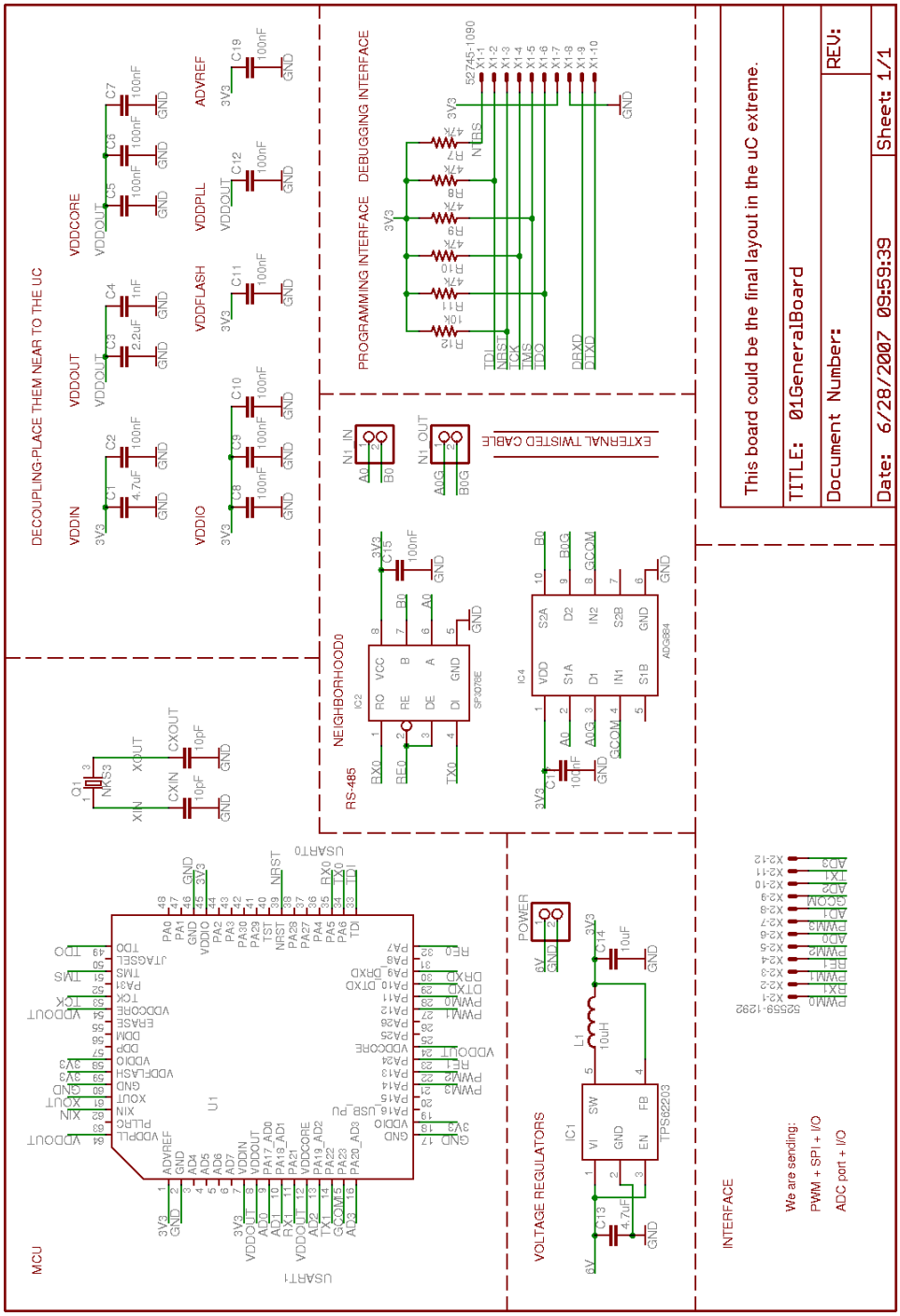


Figure B.1: Schematic diagram of the General Board - Version 1.

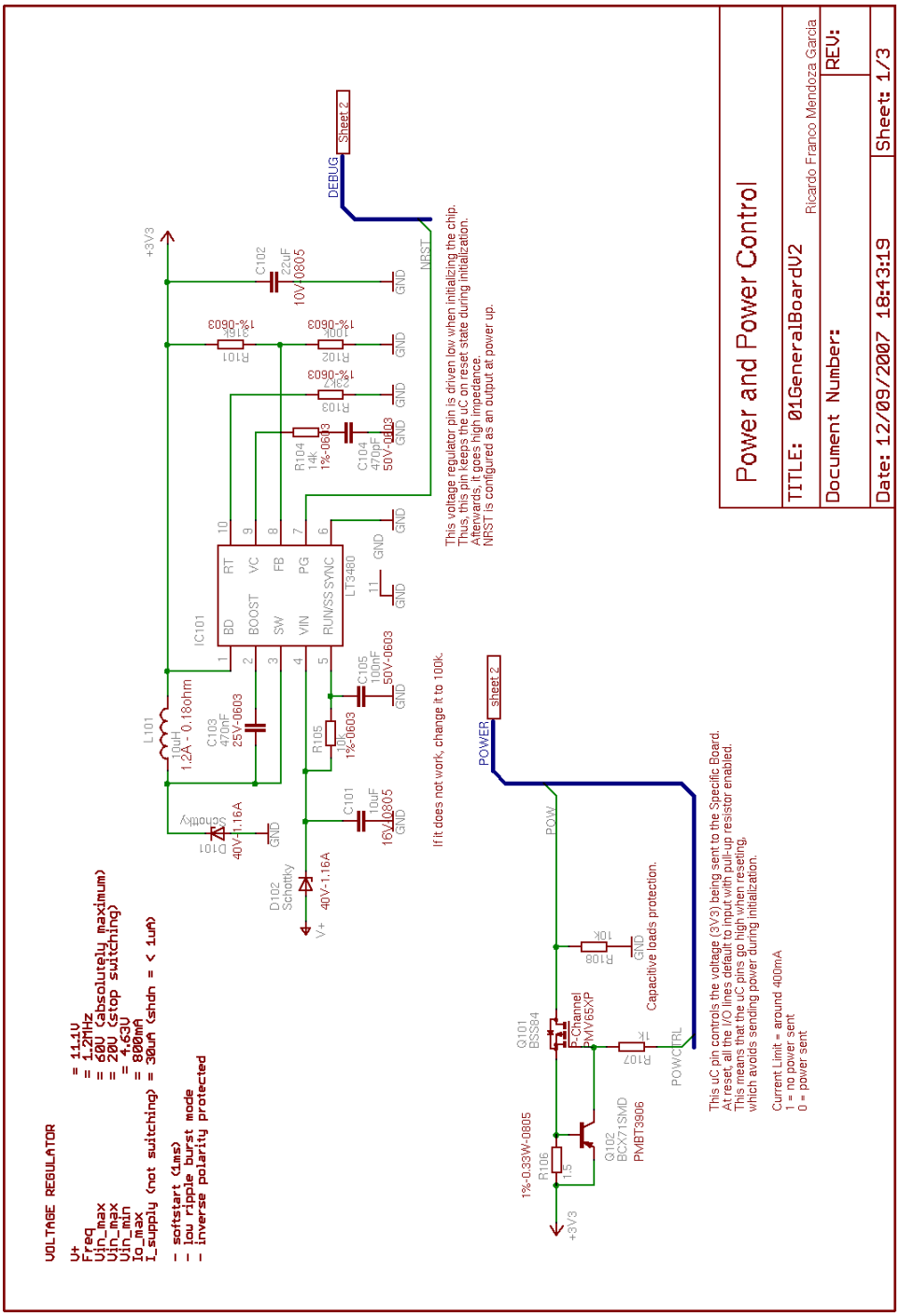


Figure B.2: Schematic diagram of the General Board - Version 2, Page 1.

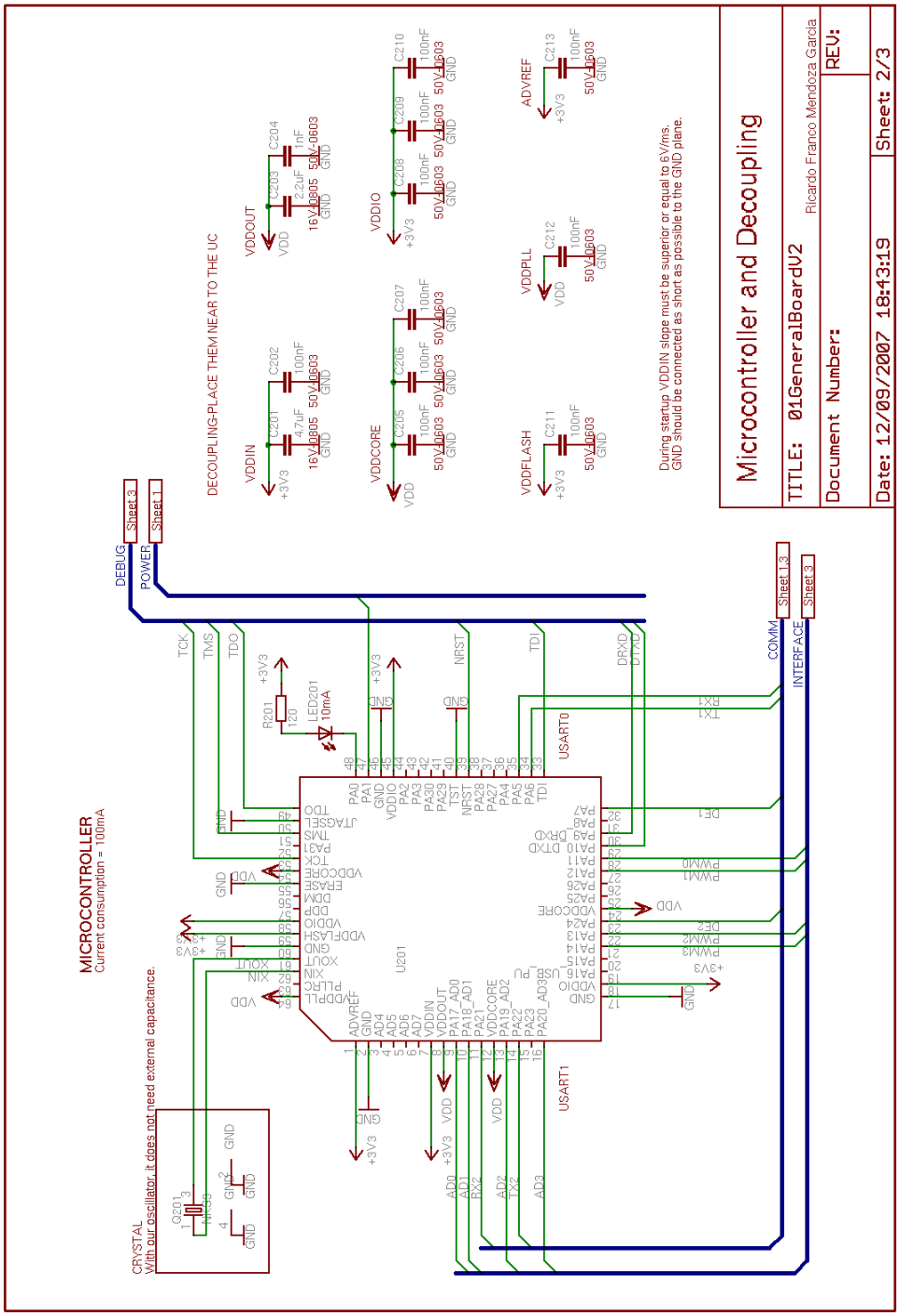


Figure B.3: Schematic diagram of the General Board - Version 2, Page 2.

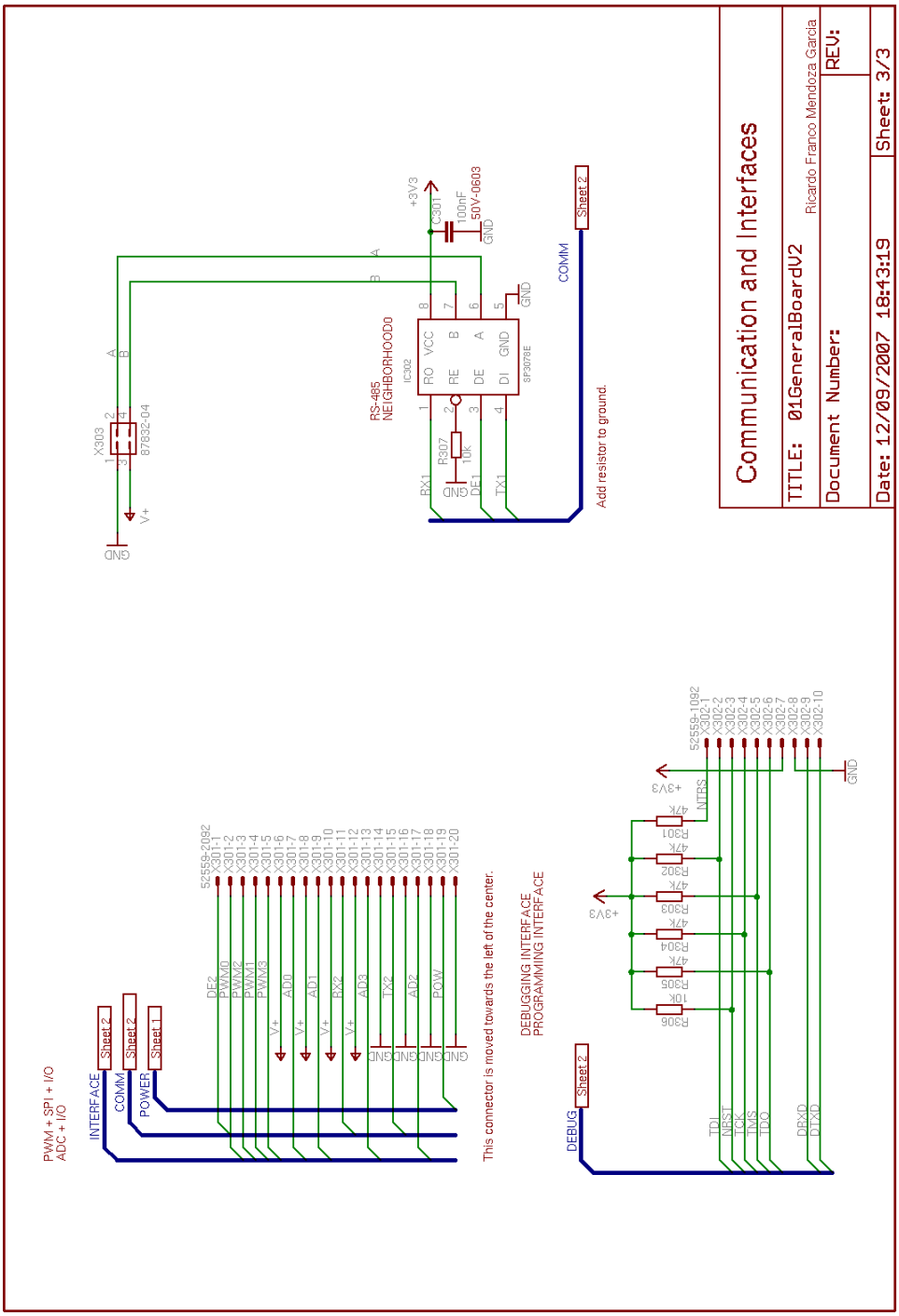


Figure B.4: Schematic diagram of the General Board - Version 2, Page 3.

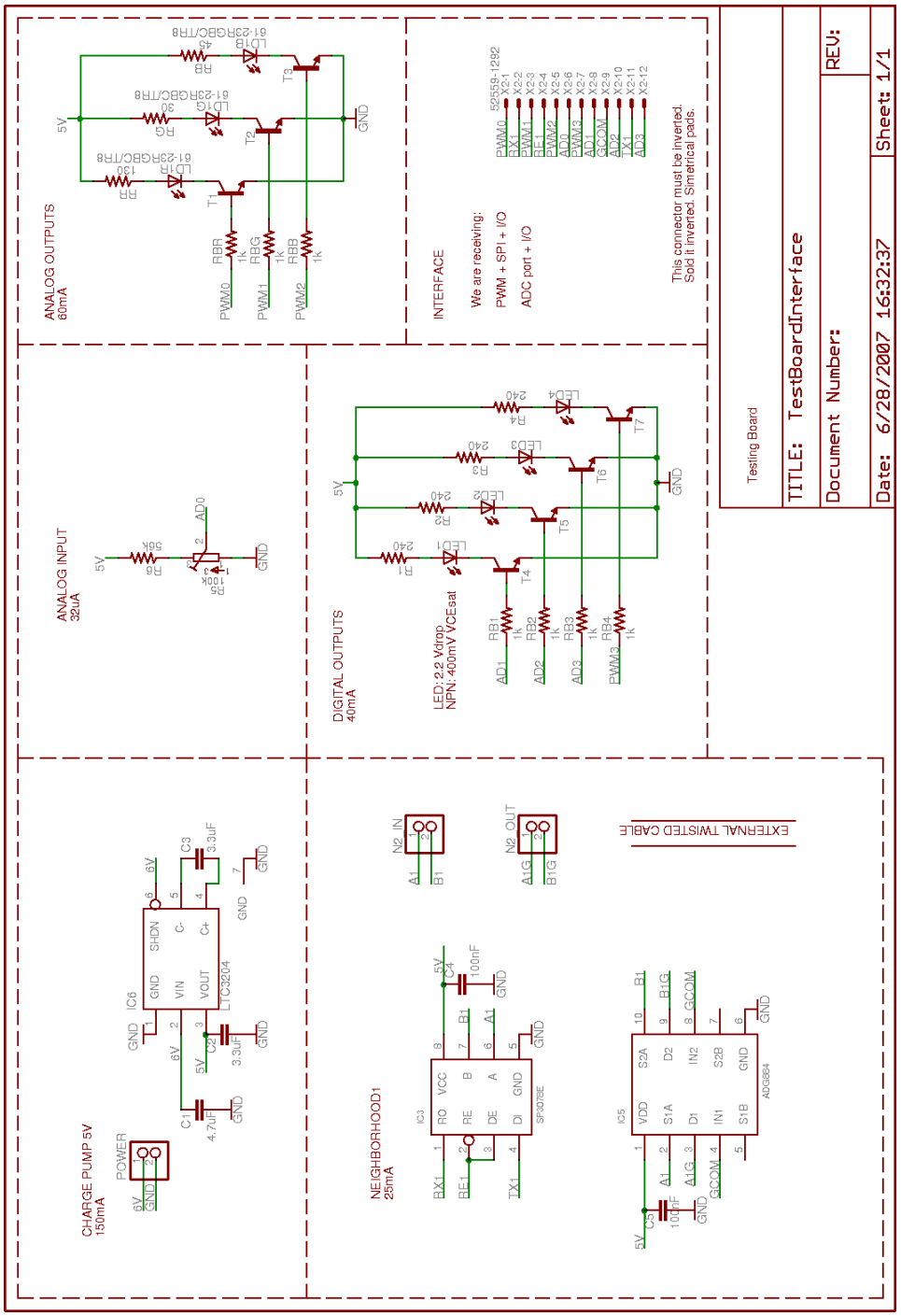


Figure B.5: Schematic diagram of the Testing Board.

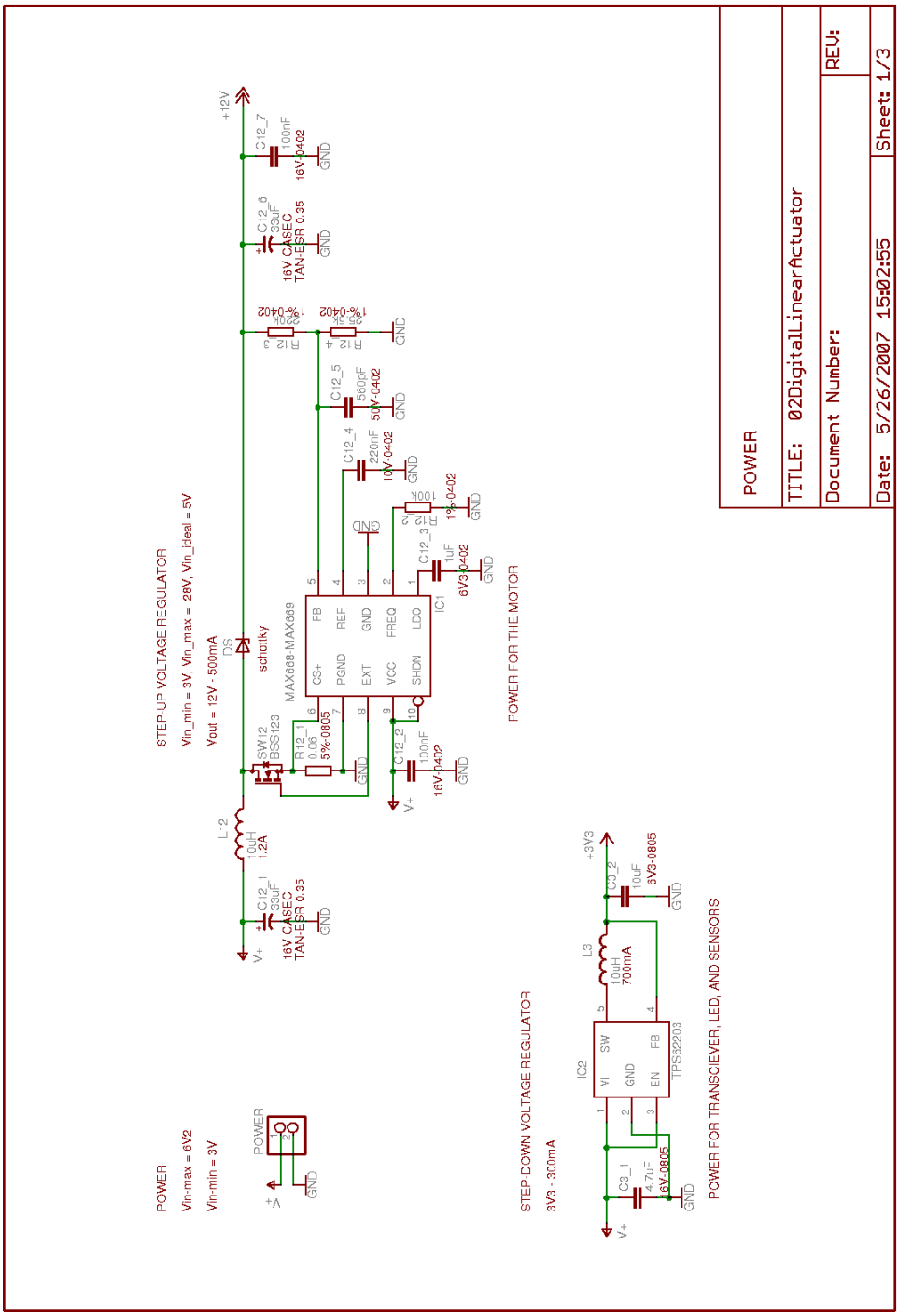


Figure B.6: Schematic diagram of the Actuator Board - Version 1, Page 1.

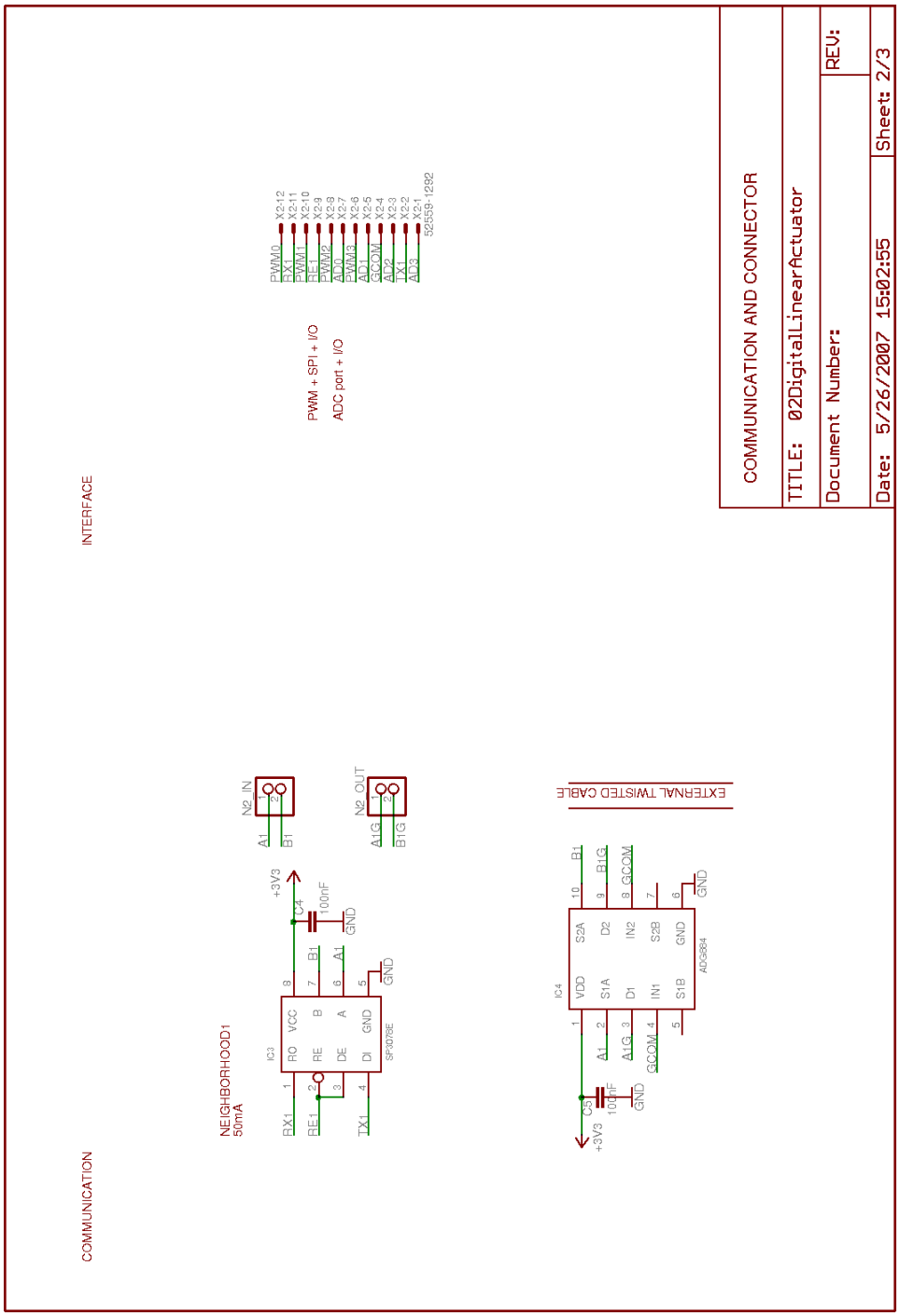


Figure B.7: Schematic diagram of the Actuator Board - Version 1, Page 2.

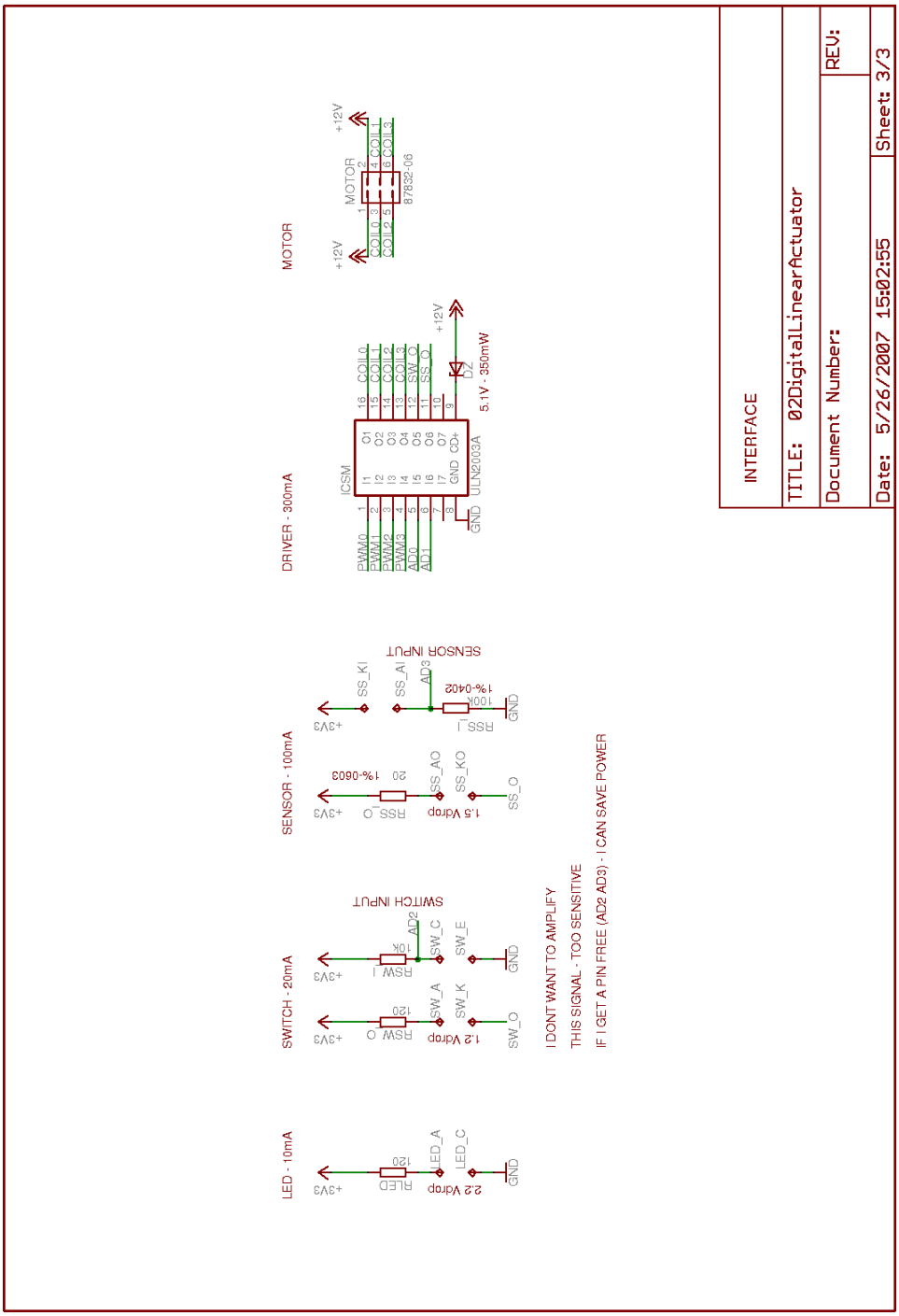


Figure B.8: Schematic diagram of the Actuator Board - Version 1, Page 3.

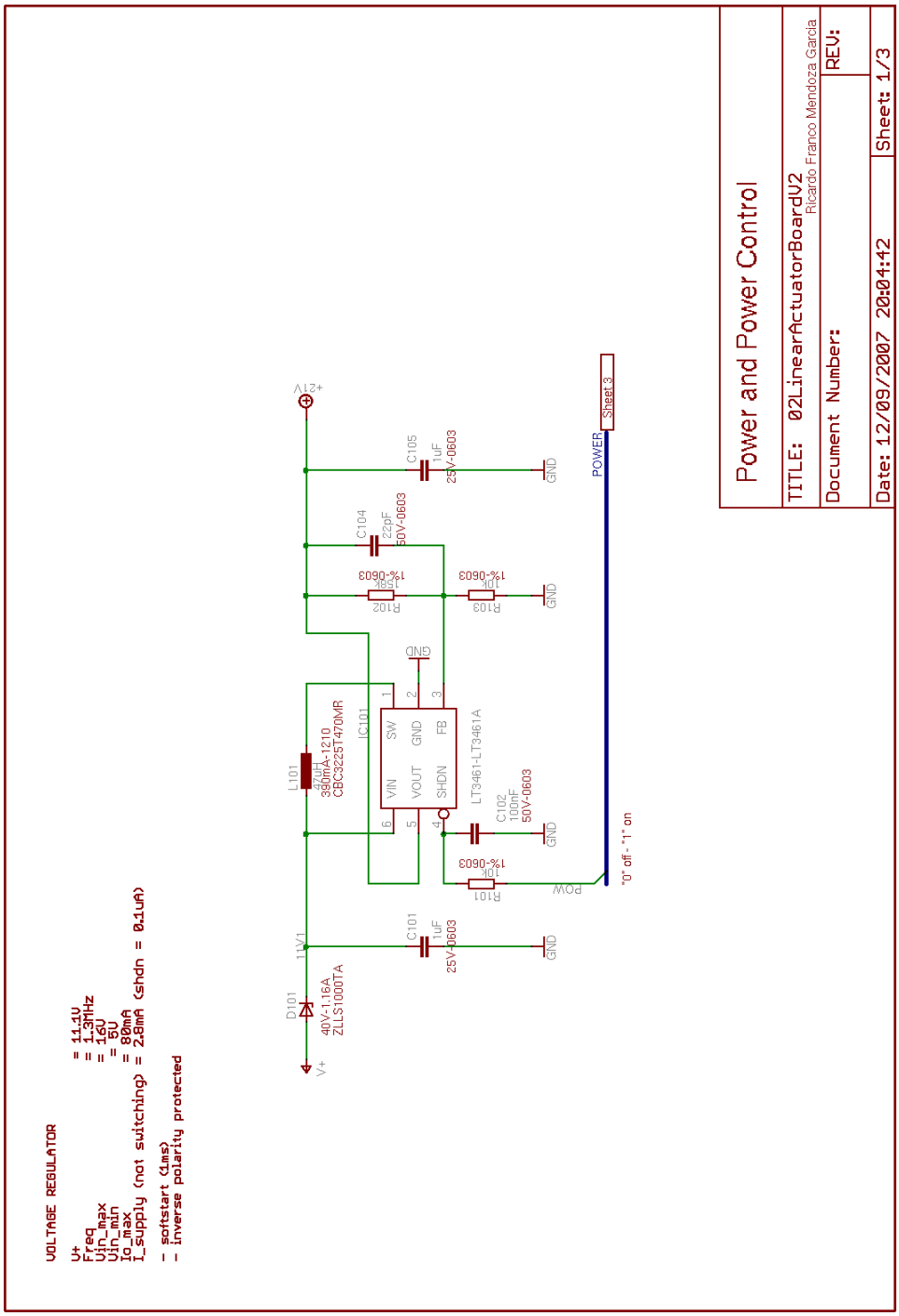


Figure B.9: Schematic diagram of the Actuator Board - Version 2, Page 1.

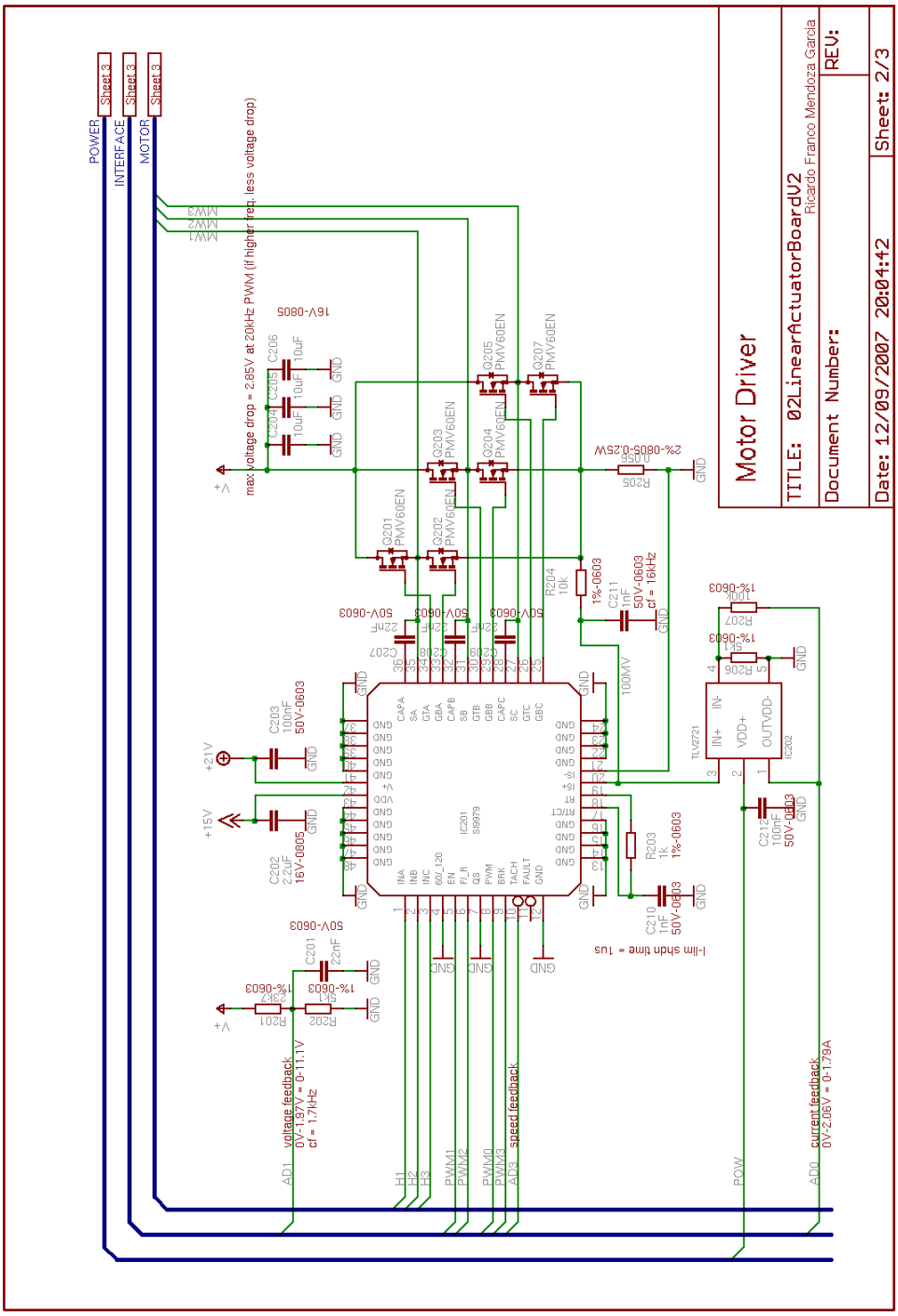


Figure B.10: Schematic diagram of the Actuator Board - Version 2, Page 2.

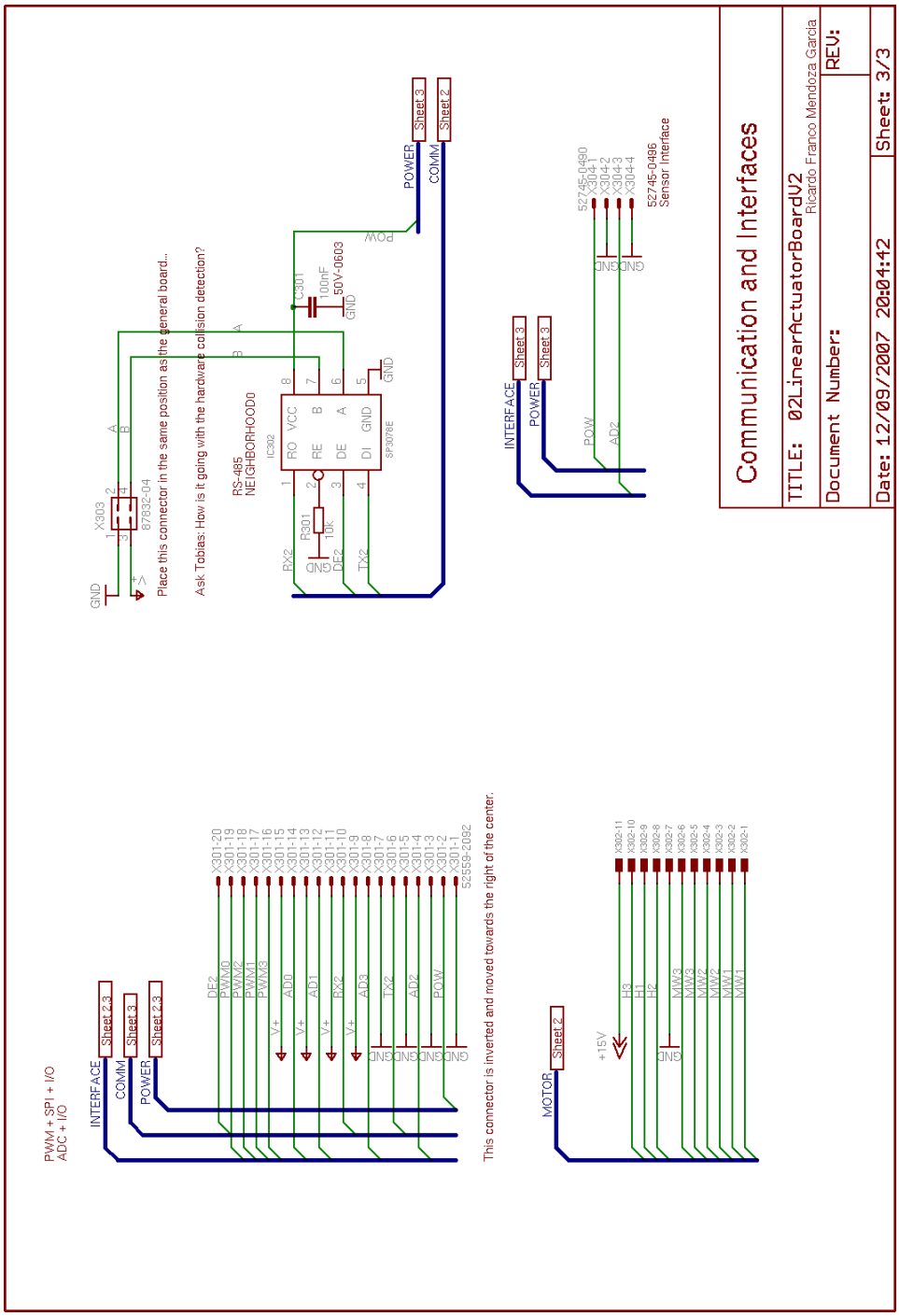


Figure B.11: Schematic diagram of the Actuator Board - Version 2, Page 3.

Bibliography

[1] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectrum*, vol. 39, no. 2, pp. 30–34, February 2002.

[2] K. Kotay, "Self-reconfiguring robots: designs, algorithms, and applications," Ph.D. dissertation, Dartmouth College, 2003.

[3] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji, "Self-reconfigurable modular robot (M-TRAN) and its motion design," in *Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'02)*, Singapore, December 2002.

[4] B. Salemi, M. Moll, and W.-M. Shen, "SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, October 2006.

[5] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems: Challenges and opportunities for the future," *IEEE Robotics and Automation Magazine*, pp. 2–11, March 2007.

[6] 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*, vol. 18, no. 5, pp. 700–712, October 2002.

[7] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji, "Automatic locomotion design and experiments for a modular robotic system," in *Transactions On Mechatronics*, vol. 10, no. 3, June 2005.

[8] H. Kurokawa, E. Yoshida, A. Kamimura, S. Murata, and S. Kokaji, "Self-reconfigurable M-TRAN structures and walker generation," in *Robotics and Autonomous Systems*, no. 54, 2006, pp. 142–149.

[9] K. Stoy, A. Lyder, R. F. M. Garcia, and D. J. Christensen, "Hierarchical robots," in *IROS Workshop on Self-Reconfigurable Robots/Systems and Applications*, San Diego CA, 2007.

[10] M. Nilsson, "Essential properties of connectors for self-reconfiguring modular robots," in *Proc. Workshop on Reconfigurable Robots, IEEE Int. Conf. Robotics and Automation (ICRA'01)*, Seoul, South Korea, May 2001.

[11] T. Fukuda, Y. Kawauchi, and M. Buss, "Communication method of cellular robotics CEBOT as a selforganizing robotic system," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, September 1989.

[12] K. Stoy, W.-M. Shen, and P. Will, "On the use of sensors in self-reconfigurable robots," in *7th international conference on simulation of adaptive behavior SAB'02*, Edinburgh, UK, August 2002, pp. 48–57.

[13] A. Lyder, "Deformable modular robots: A new direction in modular robotics," Master's thesis, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark, 2007.

[14] D. E. Ingber, "The architecture of life," *Scientific American*, pp. 48–57, January 1998.

[15] D. G. Duff, M. Yim, and K. Roufas, "Evolution of polybot: A modular reconfigurable robot," in *Proceedings, Harmonic Drive International Symposium*, Nagano, Japan, 2001.

[16] Y. Zhang, K. Roufas, M. Yim, and C. Eldershaw, "Massively distributed control nets for modular self-reconfigurable robots," in *Proceedings, AAAI Spring Symposium on Intelligent Distributed and Embedded Systems*, Palo Alto, CA, USA, 2002, pp. 45–50.

[17] Y. Zhang, K. Roufas, C. Eldershaw, M. Yim, and D. Duff, "Sensor computation in modular self-reconfigurable robots," in *Experimental Robotics VIII, Proceedings of the 8th International Symposium ISER02*, ser. Springer Tracts in Advanced Robotics, B. Siciliano, Ed., vol. 5. Springer, 2003.

[18] H. H. Lund, R. Beck, and L. Dalgaard, "ATRON hardware modules for self-reconfigurable robotics," in *Proceedings of the 10th International Symposium on Artificial Life and Robotics (AROB'10)*, Sugisaka and Takaga, Eds., Oita, Japan, 2005.

[19] A. Group, "The atron module: Fact sheet," University of Southern Denmark, Tech. Rep., 2004.

[20] D. Christensen, D. Brandt, U. Schultz, and K. Stoy, "Neighbor detection and crosstalk elimination in self-reconfigurable robots," in *Proceedings of the First International Conference on Robot Communication and Coordination (ROBOCOMM2007)*, October 2007, p. to appear.

[21] A. Castano, R. Chokkalingam, and P. Will, "Autonomous and self-sufficient CONRO modules for reconfigurable robots," in *Proceedings, 5th Int. Symposium on Distributed Autonomous Robotic Systems (DARS'00)*, Knoxville, Texas, USA, 2000, pp. 155–164.

[22] R. J. Tilley, *Understanding Solids: The Science of Materials.* John Wiley and Sons Ltd, 2004.

[23] R. F. M. Garcia, K. Stoy, D. J. Christensen, and A. Lyder, "A self-reconfigurable communication network for modular robots," in *Proceedings of the First International Conference on Robot Communication and Coordination (ROBOCOMM2007)*, October 2007, p. to appear.

[24] K. Stoy, W.-M. Shen, and P. Will, "A simple approach to the control of locomotion in self-reconfigurable robots," *Robotics and Autonomous Systems*, p. (to appear), 2003.

[25] R. Nagpal, "Organizing a global coordinate system from local information on an amorphous computer," *AI Memo*, no. 1666, August 1999.

[26] E. Yoshida, T. Arai, M. Yamamoto, J. Ota, and D. Kurabayashi, "Evaluating the efficiency of local and global communication in distributed mobile robotic systems," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'96)*, Osaka, Japan, November 1996, pp. 1661–1666.

[27] M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'2004)*, Sendai, Japan, September 2004, pp. 2068–2073.

[28] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. Ijspeert, "Yamor and bluemove - an autonomous modular robot with bluetooth interface for exploring adaptive locomotion," *Proceedings of International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR'2005)*, pp. 685–692, 2005.

[29] D. K. Cheng, *Field and Wave Electromagnetics*, 2nd ed. Addison Wesley, 1989.

[30] T. I. Incorporated, "The RS-485 unit load and maximum number of bus connections," Analog Applications Journal, 2004-Q1. [Online]. Available: http://focus.ti.com/lit/an/slyt086/slyt086.pdf

[31] J. Vo, "A comparison of differential termination techniques," National Semiconductor, Application Notes, 1993. [Online]. Available: http://www.national.com/an/AN/AN-903.pdf

[32] Maxim/Dallas, "Guidelines for proper wiring of an rs-485 (tia/eia-485-a) network," Application Notes, Jul. 2001. [Online]. Available: http://pdfserv.maxim-ic.com/en/an/AN763.pdf

[33] D. J. Christensen, D. Brandt, and K. Stoy, "Modular self-reconfigurable robots with the anatomy of animals," *The International Journal of Robotics Research*, submitted for publication.

[34] R. E. Matick, *Transmission Lines for Digital and Communication Networks: An Introduction to Transmission Lines, High-frequency and High-speed Pulse Characteristics and Applications.* McGraw-Hill Education, 1969.

[35] G. Breed, "Analyzing signals using the eye diagram," *High Frequency Electronics*, vol. 4, no. 11, pp. 50–53, November 2005. [Online]. Available: http://www.highfrequencyelectronics.com

[36] K. Stoy, "The deformatron robot: a biologically inspired homogeneous modular robot," *Proceedings of the 2006 IEEE Conference on Robotics and Automation*, pp. 2527–2531, May 2006.

[37] I. Boblan, R. Bannasch, H. Schwenk, F. Prietzel, L. Miertsch, and A. Schulz, "A human-like robot hand and arm with fluidic muscles: Biologically inspired construction and functionality." *Embodied Artificial Intelligence*, pp. 160–179, 2003.

[38] D. Christensen, U. P. Schultz, D. Brandt, and K. Stoy, "A unified simulator for self-reconfigurable robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2008)*, submitted for publication.

[39] T. Arai, E. Yoshida, and J. Ota, "Information diffusion by local communication of multiple mobile robots," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Le Touquet, France, 1993, pp. 535–540.

[40] E. Yoshida, M. Yamamoto, T. Arai, J. Ota, and D. Kurabayashi, "A design method of local communication area in multiple mobile robot systems," in *Proc. IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995, pp. 2567–2572.

[41] ——, "A design method of local communication range in multiple mobile robot system," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95)*, vol. 2, Pittsburgh, USA, 1995, pp. 274–279.

[42] E. Yoshida, T. Arai, M. Yamamoto, , and J. Ota, "Local communication of multiple mobile robots: Design of optimal communication area for cooperative tasks," in *Journal of Robotic Systems*, vol. 15, no. 7, 1998, pp. 407–419.

[43] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction To Probability*, 1st ed., A. Scientific, Ed. Addison Wesley, June 2002.