Proyecto Final de Carrera – Máster de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona
Universidad Politécnica de Cataluña

Theory and Synthesis of the Imbalance Organism

Paul Olivier

(Tutor: Juan Manuel Moreno Aróstegui)

6 January 2010 (Introductory Edition)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Presentation

This document presents the final project (PFC) for the Master in Electronic Engineering done at the Technical University of Catalonia (UPC), Barcelona. The project is an initiative into a new area of Research and Development that falls primarily under the field of Robotics.

1.2 Objectives

The long term goal of this initiative is to establish a technology that will enable the development of practical and useful nonbiological entities for the real world. The reasoning behind the specific word choice used here (e.g., practical, useful, nonbiological) will be outlined in this document.

Specific word choice is extremely important within this document and, as such, the terminology used is chosen to convey a specific meaning and done so for the sake of clarity. Clarity is of the utmost importance when it comes to Robotics. As what is common knowledge in engineering, it is difficult to achieve success if one is not clear about what one is aiming at, what the objectives are, what the problem is that needs to be solved. As well, it is imperative to compare apples with apples. When diving into a world as complex as the biological, clarity is a prerequisite before we can start to argue and compare. Hence, the reader will find below that much time is spent on defining as clearly as possible the terminology used.

This project brings to a conclusion the first phase of the long term goal. In other words, this document presents the work that has been done from when the first concepts took shape until the present day (a period of roughly two years). With respect to time scales and effort, it is predicted that the majority of the work still lies ahead.

This document therefore strives to introduce the reader to the concepts resulting from the work that has been done so far. These concepts will likely form part of the eventual foundation for a theory that is to become the driving force behind the intended technology and which I call the Theory of the Imbalance Organism (hereinafter called the IMORGal theory, of simply the theory). For a more compact representation, I will use the form IMORGal to refer to the concept of an IMbalance ORGanism.

I say 'likely' because, although I am indeed able to defend (that is, support and explain) the concepts presented in this document, I cannot guarantee they will remain completely unchanged when the foundation of the theory is established at some point in the future. This is because, as with biological organisms, the survival of this theory depends on its ability to evolve beyond threats and changes in the academic environment. The word 'survival' is significant because it is not yet clear to me that it will be possible to reach the goal of establishing the intended technology. I find the initial concept (the Need theory, explained below) extremely powerful, yet I cannot guarantee what may come of it. The idea is to find out.

Much work remains (several years most likely) before I will be satisfied enough to call the then-present body of work the Foundation for the Theory of the Imbalance Organism. This foundation will as a minimum allow the design and construction of practical and useful robotic systems for the real world, as discussed below. 'Robotic systems' is here meant in the generally accepted sense of what a robot is – it is used as a substitute until I define the term 'synthetic organism' below.

There is no doubt that synthesis plays a fundamental role in validating this theory. In my experience so far, synthesis has proven to be an invaluable tool for keeping the theory rooted in reality. Synthesis is here meant to imply simply the construction of a physical system. In fact, one of the main arguments of the theory is that a lack of adequate synthesis leads to indecisive validation of Robotics concepts and never-ending contemplation for the sake of science.

The IMORGal theory is essentially the result of rethinking all the concepts related to Robotics and Artificial Life. It reevaluates the way we look at all-too-familiar and all-too-vague expressions and terminology. As what is true of every revolutionary idea, the theory presented in this document will require people to think differently. To change one's way of thinking about the world, one's way of understanding the world, is a frightening experience which most are unable to bear at first. It shakes people's belief systems, their sense of security. Therefore, I would like to sincerely request the reader to spend the extra effort in trying to understand the thought processes that went into the formulation of the theory as it currently stands, and bear with me if any phrase sounds ridiculous, farfetched or confusing at first. I have spent a considerable amount of effort in trying to keep the text clear and simple with a layout that should flow naturally from one section to the other.

1.3 Structure

It is a daunting task to try and explain concisely and accurately the wide range of concepts I introduce in this document. Every statement made could lead to several interrelated paragraphs of explanatory nature.

The first section discusses the current concepts regarding the IMORGal theory. The first subsection (Key Description) attempts to make general statements that intend, more than anything, to start with a clear understanding of basic concepts. The following subsection (Talking Robots) discusses mainly comments from Robotics experts about the state of nonindustrial Robotics and what is still lacking in terms of making robots truly a reality in our daily lives. The subsection thereafter (Industrial Robotics Technology for Nonindustrial Robotics) discusses the strong influence that Industrial Robotics has had and continues to have on other Robotics fields. The subsection called Alternative Robotics Technologies discusses some of the approaches followed to realise a Robotics technology with a different base than Industrial Robotics. Finally, in the subsection The IMORGal Theory Speaks, several topics such as the Need theory, autonomy and adaptability are discussed as part of the IMORGal theory.

The section thereafter deals with IMORGal Synthesis. Here we go from theory to designing a physical synthetic brain for a real physical robot. The first subsection (called Design Methodology) mentions an approach that is currently followed for designing and realising a synthetic brain design. The design part consists of performing an abstract design (subsection Abstract Design) followed by the realisation part where the abstract model is physically implemented on a specific hardware device (subsection Physical Realisation). The section concludes with several remarks concerning the current synthesis process (subsection Final Remarks).

The last section (The Future) discusses the short term goals and work being done at the moment, and provides as well a general roadmap or indication as to what approach should be followed to increment the IMORGal technology complexity.

2. IMORGAL THEORY

2.1 Key Description

IMORGal Theory Statement: "Although we have been able to successfully build a vast amount and variety of robotic systems, the most difficult challenge is still that of building robotic systems that exhibit biologically comparable behaviour of a significant level."

Immediately I have to start by saying that not everyone shares this opinion about the need for comparison with biology. To some extent I agree – it depends on what problem you are trying to solve. Industrial Robotics is the best example. I will comment on Industrial Robotics further below.

The phrase "biologically comparable" implies the following:

- In terms of comparing a robotic system and a biological organism regarding behaviour, the first question that the IMORGal theory asks is that of the reason behind a specific behaviour. What is the driving force behind initiating and performing a specific action? If the robotic system moves forward, or lifts an arm/manipulator, or turns, or stops – why did it perform this action? Was it done for the same reason than that of a comparable biological organism?
- I include any robotic system where some form of comparison is made between the robotic system and biology, whether explicitly or inferred. 'Explicitly' is for example the robotic seal pet [1] or any implementation of animal locomotion [2]. 'Inferred' refers to those systems where there is a resemblance to a biological organism but the objective or intention of the system's design is not to implement specific biological behaviour. Autonomous robots, with autonomy not being a specific behaviour, could be designed with a variety of technologies or methodologies and is normally not designed to specifically behave as a certain biological organism (many exceptions to this do exist [2]). However, autonomy (which can also be called independence) forms part of the foundation for the behaviour of biological organisms.
- I include any robotic system that attempts to mimic or simulate biological movement. Examples of mimicking movement are the swimming motion of a penguin [3] and the auditory localisation behaviour of a cricket [4].
- I include any robotic device whose operation includes interaction with a biological organism. Immediately that robotic device is comparable to biological organisms simply by having entered the world of the biological organism (that is, the real world, or natural world). After all, we do not wish to interact with an alien race – the most valuable interaction is that which feels natural to us, where we can understand the robotic device's behaviour and make sense of its intentions, and where we can communicate with it (communication is meant here in the sense of the transfer of information between organisms). Taking biological organisms out of the picture might still lead to comparison, for example, a mobile vehicle meant for traversing outdoor terrain during space exploration. In such a case the comparison is made between the vehicle's mechanisms for realising autonomous operation and adapting to its environment and those of any free-moving biological organism.

NOTE: Below the use of the term "biologically comparable behaviour" implies "biologically comparable behaviour of a significant level".

According to the IMORGal theory, those who desire their robotic systems to exhibit some element of biologicalness are further divided into two groups:

- Those whose processes for generating biologically comparable behaviour are not based on biology
- Those whose processes for generating biologically comparable behaviour are based on biology

NOTE: If I classify the source of behaviour for a specific robotic system or technology as not based on biology, this does not necessarily imply a negative connotation. There are many examples of excellent work being done, regardless of the extent to which I disagree with the approach being followed for generating behaviour. A good example is the salamander walking/swimming motion [2] by Auke Ijspeert from the EPFL – note that this research focuses mostly on Central Pattern Generators (CPG) inside the brain. Additionally, the impressive work of Barbara Webb [4] on using a robot cricket to generate behaviour that is comparable to a real cricket is of direct relevance to the work described in this document (although not discussed in detail here; future work is likely to make more detailed reference about Webb's research).

The most important aspect of the division into two groups above is obviously what is meant by “based on biology” or “biology-based”. Note that it is not the physical robotic system that must be biology-based – the focus is only on the robotic system's behaviour and, specifically, the processes that generate it and influence it. Neither is it about copying biology – the goal is not to synthesise biological processes directly (for example, connecting living neurons together [5]) but to identify the abstract fundamentals that can be synthesised using nonbiological materials in order to yield nonbiological organisms with a different purpose in life than a biological organism – a purpose that will be useful to us.

An additional important aspect is the question of why there is so much emphasis on whether a robotic system is biology-based or not. The IMORGal theory says that biology is so extremely complex and optimised for efficient functioning that any attempt at generating biologically comparable behaviour (here I must emphasise: of a significant level; significantly complex behaviour) that is not based on biology will lead to ultimately inefficient and unsatisfactory results. Such attempts are merely trying to model in vain a much superior technology using inferior tools. If the desired performance of the system being designed requires only a simplified and generalised model of biology, or if biology is only intended to serve as a source of inspiration, then it might be possible to achieve satisfying results using nonbiology-based fundamentals. I suspect this will only ever be the case where there is a low level of complexity required regarding the autonomy and adaptability of the robotic device.

However, the IMORGal technology is not about small, simple robotic systems with a few specific biologically comparable behaviours – the IMORGal technology aims at using biology as a reference design to yield a single unified technology for building systems with behaviour that is comparable to bacteria, up to systems with behaviour comparable to human beings. The IMORGal technology is inherently about the synthesis of nonbiological organisms of which even the simplest organism will contain a significant level of complexity. Therefore, there is no room for adding inefficient elements to these designs – such elements will only serve to severely limit the scalability of the IMORGal technology. Scalability is a key indicator of the potential of any technology to generate biologically comparable behaviour.

Thus, after having spent astronomical amounts of resources on Robotics research over the last roughly 50 years, we still have not solved how to create biologically comparable behaviour of significant complexity. The IMORGal theory cannot prove this statement but lists a few supporting arguments (explained in detail further below):

- Research done in the field of robotics still views biologically comparable behaviour as a future goal. Note that, in general, terms such as autonomy and adaptability are used instead of "biologically comparable behaviour".
- Experts in the field still talk about some X-factor, some missing concept or technology that was not predicted, thereby affirming indirectly that current robotics technologies are not the conclusive solution. These experts additionally confirm that we are still a significant distance away from biologically comparable behaviour.

The IMORGal theory provides possible reasons for why current attempts to generate biologically comparable behaviour have been unsatisfactory:

- The extreme complexity and optimised functioning of biological systems is seen as an important possible reason. An incorrect understanding of biology leads to inefficient robotics technologies with the primary effect being a limitation on upward scalability (increasing the complexity of a simple system – for example, Evolutionary Robotics (discussed below)) or downward scalability (applying the system's concepts to the designing of simpler systems – for example, Neurorobotics (discussed below)). These robotics technologies are basically research-oriented.
- Since the envisioned robotic system must be a unique product, as opposed to simply a biological replica, the typical notion is to avoid looking too closely at biology, unless the robotic system is specifically to be used for biological research (for example, Biorobotics and Neurorobotics).
- The success of Industrial Robotics has naturally lead to attempts at generating biologically comparable behaviour using Industrial Robotics technology as the basis. This is especially the case in fields such as visual perception, navigation, control of behaviour and manipulation where these fields are developed specifically for improved robotics performance in nonindustrial locations. Industrial Robotics technology directly opposes generating biologically comparable behaviour because Industrial Robotics technology is about a robotic system functioning in a controlled specific environment performing a controlled specific task. For biologically comparable behaviour we need a technology that has biology-based autonomy and adaptability as fundamentals.
- A significant level of complexity is already required for an entry level robotic system that is autonomous and adaptable to a significant level, whether biology-based or based on Industrial Robotics technology.

In the next section I provide some information to support the IMORGal theory's claims.

2.2 Talking Robots

On the Internet there exists an invaluable resource on Robotics in the form of a podcast called Talking Robots [6]. This podcast ran until 2008 and featured 45 interviews with high-profile professionals in the fields of Robotics and Artificial Intelligence (AI). Continuing from this initiative there is a separate podcast simply called Robots [7] which is currently active. From these interviews one can get an expert view on (mainly) research that is being done in nonindustrial Robotics and AI. The Talking Robots podcast was the initiative of well-known Dario Floreano from the Laboratory of Intelligent Systems (LIS) at the EPFL in Switzerland.

Below I show and discuss selected extracts from several interviews. Two interviewers were featured in the podcasts: Markus Waibel (MW) and Sabine Hauert (SH).

Owen Holland is a professor at the University of Essex in the UK and is known for work done in biologically-inspired Behaviour-Based Robotics.

MW: "And where do you think robotics is headed in general in the next twenty years?"

OH: “Out of doors. I think we've got to show that we understand the technology well enough to produce systems that will actually cope with the real world. Not toy real worlds like a university laboratory, but actually out there in the mud and the rain, in the dynamic uncertain environment.”

MW: “And the biggest challenges in this case would be, in your opinion?”

OH: “I don't think people can complain about not having computational resources anymore. That's a problem that Moore's Law has solved. And also, I think, particularly the developments in Japan have shown that the engineering is almost there, it is almost good enough. Sensors are still a problem but on the other hand we are very highly visual creatures and the technology of visual sensing (cameras and so on) I think that is pretty well good enough. I think the problems are really in our ability to program, in our ability to conceive the correct architecture. I think the main obstacles to progress are not, if you like, not having the technology at the moment; it is not having the right ideas. I see that as the big obstacle now.”

MW: “And if you have to make a prediction for the next twenty years, where do you think the biggest advances will be made?”

OW: “I think they will be made in architectures because I think everything else is going to be incremental; we are going to have better batteries, we'll have better motors, we'll have better cameras, and so on. But I do think that the main prospects for a breakthrough are going to be conceptual.”

Thus, Owen Holland appear to indicate that there is still some fundamental concept or architecture, some breakthrough missing in order to build robotic systems that by nature can function usefully in the real world. The real world is described with terms such as 'dynamic' and 'uncertain'.

The next interview we look at involved **Inman Harvey** from the Centre for Computational Neuroscience and Robotics and the Evolutionary and Adaptive Systems Group at the University of Sussex. He has a strong background in philosophy and this will be evident from his comments. Philosophy is an important part of the IMORGal theory, especially since the IMORGal theory attempts to solve the question “Why?” - why do we do what we do when we do it? What does it mean to be alive? Inman Harvey wrote an article in 2000 called “Robotics: Philosophy of Mind using a Screwdriver” [8].

SH: “What do you mean by Philosophy of Mind using a Screwdriver?”

IH: “People do robotics for all sort of motives. Sometimes they just want some practical solution to some practical problem, where building machines to do it for them, is the obvious route to take. Some people are more interested in Autonomous Robotics, where they're trying to not just have a robot that does things for you but a robot that does things for itself to some extent. And this is an extension of the dream of artificial intelligence, to try and recreate in a machine something that replicates some, hopefully many, of the properties of living creatures, humans in particular. And when you're in this business, it's sometimes tempting to get immersed in the technical details of how do you build machines that do particular things but what I and many others would suggest is one has to stand back and think at a far deeper level than that, because inescapably when you're trying to recreate something autonomous and in some sense living, the business of Artificial Life for instance, then clearly this requires you to take some considered *thought on what it is to be alive, and these are not just technical questions, these are philosophical questions and indeed for some people, including myself on many days of the week, I have different motives on different days of the week, but on many days my motives for doing robotics is basically one means for tackling this sort of question: What does it mean for something to be alive, as opposed to being merely a machine? And these are philosophical questions to do with understanding what cognition is, what* *life is. And when you are doing this, the benefits of doing this via robotics is that it focuses the mind. I mean, for philosophers talk is cheap, but when you are doing Robotics it is not good enough to just wave your arms and say “I think this is a good idea”. You have to actually try it out and see if you can get it to work. And so in this sense Robotics is a handy tool for exploring ideas about what it is to be alive, what it is to be cognitive. A robot is a puppet that one molds and shapes to try and demonstrate, test one's ideas about what is crucial to cognition. And that's the sort of sense in which I like to use the phrase 'philosophy of mind with a screwdriver'” ...*

SH: “In your abstract for the 50th anniversary of AI, you wrote: “I shall sketch the failure of the computer metaphor for the brain, the failure of the first wave of Artificial Neural Networks, the failure of Neuroscience, the unanswered questions of Behaviour-Based Robotics. Where is the juice?” So what is the juice artificial agents lack?”

IH: “The Juice is a term that Rod Brooks uses to try and identify what's missing in autonomous robots nowadays. There are various versions of this story, and I am not sure if I am giving precisely Rod Brooks', but here goes: that we can build semi-autonomous robots in some sense that can navigate across a lab, can avoid obstacles or people. So, they give a pretty convincing picture to us, the observers, that they have in some sense intentionality. We can say “this is trying to reach the door and when an obstacle got in the way it found a way around it.” We're attributing intentionality and I am quite happy with them doing that. However, there is something missing so far, that robots might do things they are explicitly programmed to do, or if we use Evolutionary Robotics, that we've evolved them to do. We have either written in the code to make them to want to reach the door or we have evolved populations of robots such that only those that succeed in reaching the door can survive. But they don't really care about it, in the sense that a living animal does. So if you trap a wild bird in a cage or indeed in your hands, it will try and find the door, find the escape out of your clutch. But it doesn't just do it as an unconcerned “O I wonder if this way gets me out”. It will actually struggle and will really care about getting out. We can talk about them suffering, having pain perhaps. But we can't do that it seems so far with any of the robots that we might have built. They might make a good shot at finding their way out of the door, out of the cage, but they don't really care one way or the other. If they fail, they fail, and tough. But there is no emotional caring going on in there. So according to this picture there is something missing with the robots we build that, if we are trying to emulate, replicate real living creatures, we need to work out what is missing and put it in there. And if you think of it as a thing that you put in, that goes sort of with the metaphor of the juice. A sort of vitalist term really, there is something essentially missing that might not be a component it might be something else, but that is what I think Rod Brooks refers to by the juice, and other people have similar concerns, not necessarily using the term the juice.”

The IMORGal theory explicitly tries to solve this missing piece of the puzzle by allowing nonbiological systems to make sense of what is going on around it, and act accordingly, instead of blindly switching between predefined states, that is, a blind execution of its software program.

Harvey additionally mentions the 1.5mm-long Nematode worm [9]. This worm is a comparatively simple creature: the hermaphrodite Nematode worm consists of 959 somatic cells, of which 302 are neurons. Yet, not much is understood about how it generates behaviour in spite of having detailed anatomical and developmental information about the complete Nematode worm.

About the future Harvey confirms the notion of something yet undiscovered:

“What I am really hoping for is something astonishing that nobody had predicted to come up. So, the really big changes I have seen in the last 20 years, something like the Internet, or Google, for instance, these have had enormous impact on a lot of the ways how some of us interact with the world and work, and were not particularly foreseen.”

In an interview with **Barbara Webb**, one of the pioneers in the field of Biorobotics, there is an interesting comment about robotics technology. Barbara Webb is a reader at the University of Edinburgh and the head of the Cricket Lab. Her work strongly concerns biological research and is based on the construction of robots that utilise computational models to emulate specific cricket behaviour. Her work shows the important role that robots can play by serving as a tool for biologists. On the future of robotics, Webb makes the following comment:

"We probably also need some kind of <u>brain technology</u> that needs to be sorted out in some way as well. We don't generally use parallel processing for example, it is not clear that we can just replace that with fast processing which is what people do now. The fact that the computer is fast makes up for the fact that it is not parallel. But those two things might not really trade off in the long run, if you are trying to make the system really work in real-time in real environments."

The IMORGal technology explicitly aims at providing a complete technology for designing nonbiological <u>nervous systems</u>.

Rolf Pfeifer is director of the Artificial Intelligence Laboratory at the University of Zurich. He is a pioneer of 'New AI' (as opposed to traditional AI) that are based on a requirement for embodiment in order to effectively solve for intelligence. Embodiment is a fundamental concept for generating biologically comparable behaviour and is present in fields such as Behaviour-Based Robotics, Biorobotics, Neurorobotics and Evolutionary Robotics. One definition for embodiment is given by Brooks [10a]:

"Embodiment: The robots have bodies and experience the world directly - their actions are part of a dynamic with the world, and the actions have immediate feedback on the robots' own sensations."

In principle the idea is that of constructing a robot and have it operate in the real world instead of performing simulations or other computer-only attempts at solving for intelligence.

MW: "How has AI developed and changed in the last fifty years?"

RP: "... In the beginning, and up to maybe the 1970's, 1980's people in Artificial Intelligence were mainly trying to develop algorithms or computer programs for various, typically, cognitive or intellectual tasks [traditional AI] ... It was very much focussed on things like ... solving abstract problems, doing mathematical proofs, language was a big topic, reasoning, logic,... While this idea, this approach has led to very interesting applications, it has not considerably contributed to <u>elucidating the mechanisms underlying intelligent behaviour</u>. So there has always been another part of the community, that was more interested in the foundations of intelligent behaviour. And those people have realised, has started realising the limitations of this approach. <u>Realising that intelligent behaviour, especially in the real world, cannot be modelled in purely computational terms.</u> Those people were then getting interested in neural networks. Neural networks was a big relief at the time. People thought: 'Ah, no, it's really parallel, it's biological'. Where initially everybody was interested in psychology, ..., later on people got more interested in biology and started working with biologists. <u>Thus, they were attracted to the idea of neural networks, which then still didn't solve the problems that they had</u>. And then, in the mid 80's, this idea of <u>embodiment</u> came along, at least in Artificial Intelligence... It was mostly <u>Rodney Brooks</u> in the mid 80's, at MIT, that said: 'Well, I you want to deal with the real world, then you need a body to interact with the real world." And that, I think, was probably the biggest change in the history of Artificial Intelligence, much bigger than neural networks, for example. And then <u>a whole new paradigm</u> started."

Talking about the next 20 years, Pfeifer says the following:

"Where is AI going to go? We are just at the beginning of trying to understand some of the things,

we are just at the beginning of starting to understand embodiment. This doesn't mean we really have a deep understanding of it. There are so many issues that are still unresolved that people sometimes gloss over.”

With 'gloss over' Pfeifer is referring to some people saying that a robot will have the intelligence of a two and a half year old child. He then goes on to mention as an example the density of touch and temperature sensors at the fingertips of a child at that age. The richness of the information generated by these sensors for the brain (which Pfeifer refers to as the raw material for the brain) plays a fundamental role in the mechanisms inside the brain that ultimately lead to intelligence. A robotic system with only a few touch sensors will therefore form a physical barrier to yielding the same intelligence than that of even a two and a half year old child. Pfeifer adds:

“We have to be aware of the fact that humanoid robots [Pfeifer mentions this as an example robotic system], even though there is a superficial resemblance to a human being, they have a completely different morphology. Morphology is not only superficial shape but morphology is also materials, positioning of sensors, kinds of sensors, kinds of actuation systems, and so on. These are extremely different in those robots. The raw material that these robots will deliver to their brains or their controllers will be very, very different from the raw material that humans deliver to their brains to form concepts. These robots may very well be able to form concepts, or make sense to talk about concepts in these robots, that these robots operate with, but they will be very different from the concepts of humans.”

Thus, an incorrect understanding of morphology and its role in generating biologically comparable behaviour must be avoided. **Rodney Brooks**, the pioneer of Behaviour-Based Robotics, has done invaluable work in recognising and determining the balance that exists between a robotic device's physical shape, types of sensors, types of actuators and neural processing capacity. A principle of the IMORGal theory is that of necessary perception (or information extraction). The IMORGal theory goes as far as to say that all information that can be extracted from a sensor, should be extracted. Brooks gives an example below in his Talking Robots interview.

As mentioned by Pfeifer above, the work done by Brooks on Behaviour-Based Robotics has led to a fundamental shift in AI. Brooks used to be Director of the Computer Science & Artificial Intelligence Laboratory at MIT for several years and has now started a new company called Heartland Robotics. Brooks was one of the founders of iRobot Corporation which produces, amongst other products, the successful Roomba vacuum cleaning robot.

MW: “... What is Behaviour-Based Robotics?”

RB: “... a very tight connection between sensing and acting, to do a very simple behaviour, and then we'll add another layer on top of that which is also connected to sensing, to action, to do an additional behaviour, and we laid those behaviours on top of each other, much as, in a Scientific American sort of article, one would talk about evolution as being new capabilities added over time to an evolving brain in some creature. So, everything was very tight between sensing and action, so we didn't need a lot of computation to deal with more complex worlds.”

“...In the early days, when there wasn't much in the way of computer power that we would have on board, we very much had to get by with minimal sort of processing, so instead of processing a whole image, we'd try and project out what were the critical things one needed in order to solve the problem. So, imagine a robot that is living in a world with flat floors, and a camera pointed slightly down on the head, and there are two objects out in the field of view of the robot. The further one away appears higher up in the image. So, rather than trying to compute the actual distance to the objects out in the world, which involved knowing a lot about the geometry of the camera and maybe a lot of processing, instead just compare the height of an object in the image

to some threshold and if it's lower in the image then some threshold, then we the designer know that that object must be pretty close by, so maybe the robot should take account of it. So, instead of not processing the whole image to figure out everything that is out in the world, you just need to look at the bottom of the image. Is there any stuff there that is inhomogeneous or does it all look like a smooth floor? If it's just a smooth floor, you can go ahead. If there is something in the bottom of the image, then you better process it and respond to it."

"...Thus it is very much about minimising the sensory processing, in order to achieve some behaviour, and insects certainly seem to have used that strategy in their evolution. And when we look at humans, we see that humans, actually humans use that strategy a lot too."

"...The projects I am really interested in right now, is getting robots to manipulate the world. The successful robots that are out commercially are all navigation machines, they navigate around and they do some task as a side effect. Maybe they vacuum the floor, maybe they plough a field, maybe they get a sensor out to some military-relevant site, but they're navigation machines. I think the real use of robots, widespread use of robots, is going to be when robots can touch the world and physically manipulate it, so we have been pushing on manipulation for our robots."

Brooks comments on some of the technology aspects that need attention:

"... But I see four research challenges for, what I call, behaviour-based robot research which is that I think as we make progress on any one of them, we'll improve the applicability of robots. And I make these four challenges in comparison to what children can do. So the first one is object recognition at the level of a two year old child. A two year old child can come into a room they have never been in before. You ask them what this thing is, and they say it's a chair. "What's this?" "It's a table." "What's this?" "It's a cup." Where they haven't seen a chair that's got that particular shape before, that particular colour. They haven't seen a table with that particular number of legs before but they can generalise and recognise objects. And home robots right now can't even really tell the difference between a person and a chair. So, object recognition capabilities of a two year old. Anything moving towards that is going to increase the discernment of the robots greatly."

"Next level is the language capabilities of a four year old child. A four year old child can listen to people in noisy environments, can deal with different accents, deal with conditional sentences and relate those words to objects in the world and carry out dialogue. That's the second challenge. The third challenge is the manual dexterity of a six year old child and I say six year old because a six year old can tie shoelaces, they can deal with floppy objects and manipulate them. And a six year old child can do pretty much manually every task that you need done in the house, and every task that a chinese factory worker does in manufacturing of the consumer products in the world. And the fourth challenge is the social understanding of an eight year old child. An eight year old child ... introduced into a home situation can recognise who the parents are, recognise the dominant hierarchy, they can recognise when someone says one thing and means another, how their actions relate to their words. And eventually we'll want our robots to understand that level of interaction. So there are the four challenges."

Brooks further mentions the concept of something missing in a different interview with the web site Edge, entitled "The Deep Question" [10b]:

"The question then is whether there is something else, besides computation, in all life processes? We need a conceptual framework such as computation that doesn't involve any new physics or chemistry, a framework that gives us a different way of thinking about the stuff that's there. Maybe this is wishful thinking, but maybe there really is something that we're missing. We see the biological systems, we see how they operate, but we don't have the right explanatory modes to explain what's going on and therefore we can't reproduce all these sorts of biological processes."

These are the missing elements that the IMORGal theory wishes to solve – a new way of thinking, of understanding the biological fundamentals of behaviour.

Before I comment on some fields of nonindustrial robotics research, which correspond most to the IMORGal technology, I have to point out several principles of Industrial Robotics technology in terms of generating behaviour. Industrial Robotics technology plays a primary role in the nonindustrial robotic products that exist already. These nonindustrial robots are successful not because they show biologically comparable behaviour but because it was possible to control the environment to such an extent that Industrial Robotics technology could be applied.

NOTE: The IMORGal technology is from the start not primarily aimed at creating a research field but rather how to realise practical and useful products with biologically comparable behaviour.

2.3 Industrial Robotics Technology for Nonindustrial Robotics

2.3.1. The Nature of Industrial Robotics

For more than two decades now, Industrial Robotics has been successfully incorporated into manufacturing, providing increased efficiency, quality and productivity. The main industries utilising these robots are motor vehicles, automotive parts, chemical, rubber and plastics, metal products, electrical machinery and electronic components. These robotic systems are characterised by operating in a controlled specific environment performing a controlled specific task. It is mostly this control that leads to the significant above-human level of efficiency and performance. Thus, this level of control is necessary.

The environment is effectively designed itself. The environmental factors are studied and modelled. Those factors that cannot be controlled are compensated for in the robot's design. Those that can be controlled are designed together with the robot. All these factors are calculated to sufficient accuracy a priori, that is, before the robot starts operation.

To further explain:

NOTE: These concepts are closely related. In a sense, the effort of making an environment or task specific, requires control.

- Controlled Specific Environment: This refers to removing uncertainty from the environment during the design process. For example, confining the robot to a specific location or area, keeping any object, other robot or human out of its way such that there is no interference with its operation, adapting to the robot's manner of operation such that we can help it perform its task (for example, placing objects to be processed in a certain order and arriving at the robot with a speed that is acceptable for the robot), or programming the robot to perform certain actions after being keyed by us (for example, when the robot should 'say' a preprogrammed phrase). Following from being in a controlled environment, the environment is further defined in specific terms. For example, indoors, a specific location in an assembly line according to the function it needs to perform, installation requirements, ensuring the required operating temperature range is maintained.
- Controlled Specific Task: This refers to those elements in the design process where we make the

task easier or simpler for the robot, giving the robot as few tasks as possible to do (preferably one). For example, transportation of the robot to and from the operational area, the operator deciding when to activate/deactivate the robot, the operator programming the robot with instructions for its execution, the robot being exposed only to the right input elements or material which it can successfully process, having an override switch or an operator nearby. Following from having a controlled task, the robot is left with nothing else to do but follow its preprogrammed design. At times the task can contain different modes or configurations – these are all tested during the design phase and operation merely consists of selecting the most appropriate mode or configuration. Additionally, the specific task is not for instance to machine something but machine a specific part of a specific material using a specific technique.

The success of Industrial Robotics has led to the familiar concept of moving robots out of the factories and into our daily lives. That is, new markets are emerging such as professional and service robotics, domestic robotics, education and entertainment robotics and security and exploration robotics. To what extent can those technologies implemented so successfully for Industrial Robotics be used as a base for these emerging markets as well? Why would it be fundamentally different to design a robot for doing a specific task than having specific abilities for performing various tasks, for a controlled specific environment than for a dynamic uncertain environment?

The problem so far has been that it is no longer possible to have a complete design that can fully compensate for (some of) these dynamic and uncertain elements. The new robotic systems should now be able to respond adequately to these elements at the moment when they are encountered during the operational life of the robotic system. That is, instead of including the solution to a specific problem based on a priori knowledge, one needs to add a mechanism or process that will enable the robot to generate a solution when one of several unknown but specific problems are encountered in the environment at a specific time.

Note that a successful design for an industrial robot is always optimised for its specific environment and task. There are no general values or general behaviours that are up to the robot to determine – every element is precise and tightly connected to the specification for the robot's environment and task. For example, it is not about controlling some robot arm but one that has a specific size, weight, degrees of freedom, number and type of motors, power consumption and so forth. Only that which is essential to the operation of the robot is included in the design, meaning, the robot is fully designed as a unit for that specific environment and task. Repeating this design process for every unique product, even when only slightly different, including future product upgrades and maintenance, implies a significant increase in cost and resources. Thus, concepts like modularity, code reuse and configurability have emerged to decrease design time and effort. However, it still stands that for a specific product there is a specific set of features and a single configuration that are tightly connected to the specific set of environmental and task features at that moment. Thus, every such product is optimised for its environment and task. It is this optimisation that enables the robot's impressive capabilities and therefore its usefulness.

An important element regarding the environmental uncertainty encountered in the emerging robotic markets is that of what the robot must do. How can we sufficiently describe its purpose given that this purpose is no longer as inherently clear as in the case of having a specific environment and task? Vague or unclear objectives are not good practice in engineering. What will the robot be busy with specifically at any given point in time? We don't want the robot to waste energy doing either nothing or something useless. Previously its task was well-defined and specific whereas now there is a focus on an ability or skill and being able to repeat the same action or purposeful behaviour within a variety of environments. In addition, by design, the Industrial Robot is typically confined to a limited space or habitat. In the emerging markets, the robots must function in a larger space and/or different kinds of spaces. Instead of dealing with tasks, we now have to deal with endowing the robot with purposeful conduct or behaviour.

Intuitively it feels though that by taking away the specificalness of the robot's task, the robot's usefulness significantly declines. It is almost as though one can imagine a picture of a robot inside a room doing nothing really specific, not being really good at anything, driving itself around almost aimlessly, without real focus, perhaps performing a few tricks for entertainment purposes, wasting energy instead of being productive doing something specific. Perhaps this picture is not fully correct and the issue of usefulness will not be a problem. Or perhaps it is indeed an issue and one needs to specifically take the robot's intended usefulness into account when thinking about futuristic robotic designs. Having the mere ability of, for example, long term memory does not make the robot inherently useful. On the other hand, having a specific task to do does seem inherently useful, regardless whether for a specific environment or not.

The extent to which the environment is unknown in the emerging robotic markets varies from application to application. For certain cases, the robotic system being designed finds itself in an environment that can be made controllable (compare the navigation-based products mentioned above by Brooks). These applications where the habitat of the robot is still much controlled, albeit in an unusual habitat compared to conventional Industrial Robotics, appear only slightly different from Industrial Robotics. This is only meant in the sense that, upon close inspection, the scene of operation of the robot may have changed but the realisation of the solution is still based on controlling the environment to the extent that the robot is required to do nothing more that a controlled specific task. Further below I will use a golf green mowing robot (the RG3 by Precise Path Robotics) as an example to make this observation clearer.

2.3.2. Nonindustrial Robotics Products

Let us first look at some typical examples of nonindustrial robotics products. Robotics Trends Publishing is *“an online news, information and analysis portal focused on business and technology trends for people who build, buy, invest in, and seek to understand the personal, service, mobile and military robotics market”* [11].

The web site divides robotics into the following areas (some example applications from the web site are listed additionally):

- Personal Robotics
  - Example Applications: Toys and robots for hobbyists, robotic gaming, household robots, personal entertainment robots, floor-scrubbing robots, educational robots, robot-assisted walking therapy, human-robot interaction research, elder-care robots, social welfare robots
- Service Robots
  - Example Applications: An emerging telepresence industry (where the robot effectively becomes your eyes and ears at a remote location, saving you the trip), ocean exploration, Unmanned Aerial Vehicles (UAV) for chemical detection during possible chemical warfare attacks, Unmanned Guided Vehicles (UGV) to serve as *“self-driving cars, soldier-following carts, fruit-picking wagons and teams of reconnaissance scouts”*, lunar rovers (that is, vehicles for the moon), *“mobile robotic solutions for warehouse automation”*, robots to *“increase productivity and efficiency in manufacturing environments”* by Heartland Robotics, Remotely Operated Vehicle (ROV) technology
- Security and Defence

  - Example Applications: All types of *“intelligent autonomous land-based and aerial unmanned systems to perform their missions in hostile or challenging environments”*
- Industrial Robots
- Academics and Research
  - Example Applications: Robotic wheelchair with self-navigation, perception for human robot interaction, detection of humans and objects in video, adaptive decision-making (under Jeff Krichmar about whom I will say more in a different section below), robot emotions, robotic hand technology, undersea communication and navigation, next-generation radar systems for robotic vehicles, surgical robotics for local and teleoperation, autonomous underwater vehicles, robots to communicate with and help the elderly with limited mobility
- Design and Development
  - Example Applications: The focus is here on tools and technologies that enable the design and development of robotic applications. These especially include mechanical components, software, sensors and actuators

NOTE: The examples above serve only as illustrations - the point is not to describe these applications in detail but to give a brief glimpse into a representative set of nonindustrial robotics. It is assumed that the reader is either familiar with the mentioned robotic application or able to conceive the basic idea behind its operation and its designed purpose.

From the robotic areas above one can draw a few conclusions.

(1) It seems the list is almost endless in terms of the areas robotics can be applied to.

(2) An application can easily span more than one area, making it sometimes difficult to decide under which section to include the article.

(3) Establishing a robotic product is not a one man task: in all the applications mentioned above there are teams spending a vast amount of time and money on providing sometimes only a subsystem of a larger robotic solution. The entry barrier appears mostly exceptionally high.

(4) It seems that most of the applications fall, in part or completely, under the concept of Service Robotics, that is, a robot that performs some service (sometimes called nonindustrial robotics [12]). This is quite natural. In the manufacturing industry it is easy to notice the advantages of using robotics over manual labour – they are more productive, more accurate, more efficient and do not get tired of repeating the same task a million times. However, in the nonindustrial world, what is it that robotics can offer us? The most obvious is that of providing a service. Hence, Service Robotics inherently sounds like something useful.

The importance of Service Robotics is confirmed by looking at the web site for the International Robot Exhibition 2009 in Tokyo [13]. The list of exhibitors are divided into only a few sections, with Industrial and Service Robotics by far being the two most important categories.

The web site of the important International Federation of Robotics (IFR) has sections for Industrial Robotics, Service Robotics and Robotics Research (the aim of this research section being to bridge the gap between research and the industry) [14]. The IFR acknowledges the difficulty in exactly defining what Service Robotics is but does however give a provisional definition:

“A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations.”

The IFR's section on Service Robotics has a Product subsection which lists the following product classification:

- Personal / Domestic Robots
  - Robots for domestic tasks
    - Robot butler/companion/assistants/humanoids
    - Vacuuming, floor cleaning
    - Lawn mowing
    - Pool cleaning
    - Window cleaning
  - Entertainment robots
    - Toy/hobby robots
    - Robot rides
    - Pool cleaning
    - Education and training
  - Handicap assistance
    - Robotized wheelchairs
    - Personal rehabilitation
    - Other assistance functions
  - Personal transportation (AGV for persons)
  - Home security & surveillance
- Professional Service Robots
  - Field robotics
    - Agriculture
    - Milking robots
    - Forestry
    - Mining systems
    - Space robots
  - Professional cleaning
    - Floor cleaning
    - Window and wall cleaning (including wall climbing robots)
    - Tank, tube and pipe cleaning
    - Hull cleaning (aircraft, vehicles, etc.)
  - Inspection and maintenance systems
    - Facilities, Plants

  - Tank, tubes and pipes and sewer
  - Other inspection and maintenance systems
- Construction and demolition
  - Nuclear demolition & dismantling
  - Other demolition systems
  - Construction support and maintenance
  - Construction
- Logistic systems
  - Courier/Mail systems
  - Factory logistics (incl. Automated Guided Vehicles for factories)
  - Cargo handling, outdoor logistics
  - Other logistics
- Medical robotics
  - Diagnostic systems
  - Robot assisted surgery or therapy
  - Rehabilitation systems
  - Other medical robots
- Defense, rescue & security applications
  - Demining robots
  - Fire and bomb fighting robots
  - Surveillance/security robots
  - Unmanned aerial vehicles
  - Unmanned ground based vehicles
- Underwater systems
- Mobile Platforms in general use
- Robot arms in general use
- Public relation robots
  - Hotel and restaurant robots
  - Mobile guidance, information robots
  - Robots in marketing
  - Others (i.e. library robots)
- Special Purpose
  - Refueling robots
  - Others

– Customized robots

– Humanoids

There exists a robotic field called Personal Robotics, which could either be seen as part of Service Robotics or a field on its own, depending on the specific application. Personal Robotics mostly has to do with entertainment, educational and therapeutic robots, for instance the Nao robot by Aldebaran Robotics [15]. As with the case of the IFR products mentioned above, Personal Robotics could also include robots for domestic use. The Robotis web site [16] provides an additional description of Personal Robotics. They refer to:

"Robots and human beings co-existing in our daily lives".

2.3.3. Example of the RG3 Green Mower

It is not always easy to see all the sometimes subtle aspects of the control we exert on the environment and task. I list here an example where it seems as though the robot requires a less controlled environment and shows a greater ability to function in a more complex environment (to a certain extent it does).

I am referring to the RG3 Golf Course Mower by Precise Path Robotics [17] which I find to be an excellent product from an engineering point of view. It is essentially an autonomous lawn mower for golf greens utilising a Local Positioning System (LPS, as opposed to GPS) consisting of several poles (beacons) placed at exact locations around the green for navigation.

The control of the environment here has mostly to do with limiting the when and where of the robot's operation. I refer, for example, to the robot being stored or contained within one space when it is not operating (storage), transported to the operational site, powered up and started to perform its task after which it is switched off and transported back to storage. When the robot is thus started, everything is in place, everything has been prepared, the green is ready for mowing, there are no golfers around, and mowing of the green can commence. The mower only operates on golf greens and not on the other grass-covered areas of the course. Any golfer will know that a green is the smoothest grass surface on the course, with the finest grass and where every small piece of rock, leave or twig is a nuisance for putting. Compared to your everyday domestic lawn with its holes, bumps, rough edges and grass of varying type and thickness, a golf green is a far greater controlled specific surface and has much less uncertainty that the robot has to deal with. On the other hand, green mowing requires a high level of performance: the grass length must be accurately cut, the cutting pattern is specific and requires almost perfectly straight lines, and no damage must be caused to the green or collar (the edge of the green) surfaces.

The specific site for performing the task may sound unusual compared to Industrial Robotics but the element of significant control is unmistakable. However, I do not mean to say that this robot lacks the ability to drive itself to the green – I am trying to point out that, although a successful and impressive product, it is still designed on the controlled environment and task principles. It turns out that golf green mowing is a specific niche in the market in which the environment can be sufficiently controlled such that a successful product does not require principles different to that of Industrial Robotics. Obviously the RG3 does require additional engineering efforts above what are typically encountered inside a manufacturing plant.

The successful nonindustrial robotic products out in the market use conventional existing engineering thinking that develops a solution based on controlling a specific environment to the extent required for yielding a robotic system that can perform some controlled specific task in a successful way. The methodology (that is, the approach on how to solve the problem) always ends up with a set of parameters to which the robot is designed: how fast it must do movement X, how long it must assert level Y. Each parameter has its own valid range or tolerance that will allow the robot to perform correctly. Each parameter is designed before the robot starts operation, including those parameters that are configurable during operation. The control theory implemented is all about ensuring that the design value of a parameter is reached, maintained or adhered to.

Is this wrong? Of course, not. The successful products are successful for a reason.

Let us look at some of the design parameters of the RG3:

- Straight line path accuracy: 2.5 cm (This refers to the ability to move in a straight line when mowing the green – an important characteristic of a high quality golf course green)
- Operating Speed: 1.5 m/s maximum
- Gentle and smooth turning to avoid damaging the turf
- Location of turning: green, collar or outside the collar
- "Rolling only" feature (the green surface is compacted but no cutting is performed)
- Mowing direction: normal and reversed directions (for straight line cutting), clockwise or counterclockwise (for the clean-up cut around the perimeter of the green)
- Scheduling of when mowing is to be performed
- Power: 3 rechargeable lead acid batteries, giving 180 minutes of operation; require 8 hours to fully charge

Now let us look at some of the control required to make this product work:

- Navigation Setup: Four beacons are placed at chosen locations around the green. Communication, based on ultrasound and infrared, is possible between the beacons and the mower. Before first operation, the mower is pushed by an operator around the perimeter of the green to 'map' the specific green. This is recorded and used for all mowing operations to follow.
- Starting operation: The mower is transported to any location near the green. The operator pushes the 'Start' button and the mowing commences operation automatically. After completion the mower returns to the starting position from where it must be transported to the next green or taken back to storage.
- Supervision: An operator is always near to check that the mower is operating correctly.
- Operational Navigation: The navigation system (based on the four beacons) is able to accurately guide the movements of the mower. The specific pattern of movement (including all configurable operational parameters) is preprogrammed onto the RG3.
- Configuration: All configuration are done via specific software that allows a supervisor to select the time of mowing, mowing pattern with turning location and speed.
- Power: An operator needs to ensure that the mower's batteries are recharged before operation.
- Maintenance: The cutting blade can be removed and sharpened however it is not clear if the mower can detect when sharpening is required (I suspect not). I would additionally suspect the mower to have several status indicators to indicate to the operator values such as current battery power level.

There are a few important remarks about these elements of control, regarding the hypothetical case of desiring to remove some or all of the control and still have the same end result:

- Every element of task control that is removed has to be replaced by an element of autonomy. Where the task control was external to the robot (done by a human), the new autonomous behaviour must be an internal capability of the robot.
- Every element of environmental control that is removed has to be replaced by an element of adaptability. Where the environmental control was external to the robot (done by a human), adaptability must be internal.
- The elements of autonomy and adaptability are always present in every design. It is simply a matter of who performs them: the human (externally) or the robot (internally). If the robot does not adapt to its environment, the environment (which includes humans) must adapt to it. This is a common frustration with Human-Machine Interfaces (HMI) – in the end it is we that need to adapt to the controls of the dumb machine. The fact that the machine has a large number of configurable options does not help much – these are mere facilities for adaptability added by humans and eventually set by humans instead of the machine.

Note the following limitation due to practical reasons regarding a decrease in environment and task control for the RG3: a green mower requires fine mechanics for precise cutting and gentle rolling – attempting to drive this mower over other surfaces (for example, a gravel road) will likely lead to mechanical damage. Additionally, giving the robot the ability to drive itself between greens adds no value to the mowing performance – it would be a different case if there were no humans nearby and the mower was forced to get from green to green autonomously (perhaps using different mechanics for nongreen driving). A further point to note is that by controlling the environment and task as has been done for the RG3, the manufacturer has kept the complexity of the RG3's design to a minimum – any element added to increase autonomy or adaptability would have caused a significant increase in system complexity. This is a concept that we will see often and clearly in the future (beyond the scope of this document): when an element of environment control is replaced by a process of adaptability, the increase in system complexity is perhaps exponential – I mean to say: much more than double. In the case of the robot's speed and the current RG3 design, the speed is set by the designer or operator – the RG3 control system simply has to execute this value. With adaptability present there would be no speed value present at first. The mechanism for establishing this value is heavily dependent on having sufficient sensory feedback. The RG3 would have to observe perhaps wheel velocity and balance sensors, and other visual information regarding the green surface, including its own state (weight, friction, motor performance levels), in order to establish a speed that takes into account all this information and allow successful mowing of the green.

If the replacement of control with autonomy and adaptability implies a significant increase in complexity, what does that say about entry level robotic systems that attempt at exhibiting biologically comparable behaviour? For one, that significant complexity will be required for implementing autonomy and adaptability even for the 'simplest' of robotic systems. However, even more important, that the technology must be super efficient – there is no 'space' for designs that have a poor system-complexity-to-behaviour-complexity ratio.

Figure 1 shows a spectrum, which I call the Control Spectrum of Behaviour, which highlights the trade-off between control and autonomy/adaptability mentioned above.

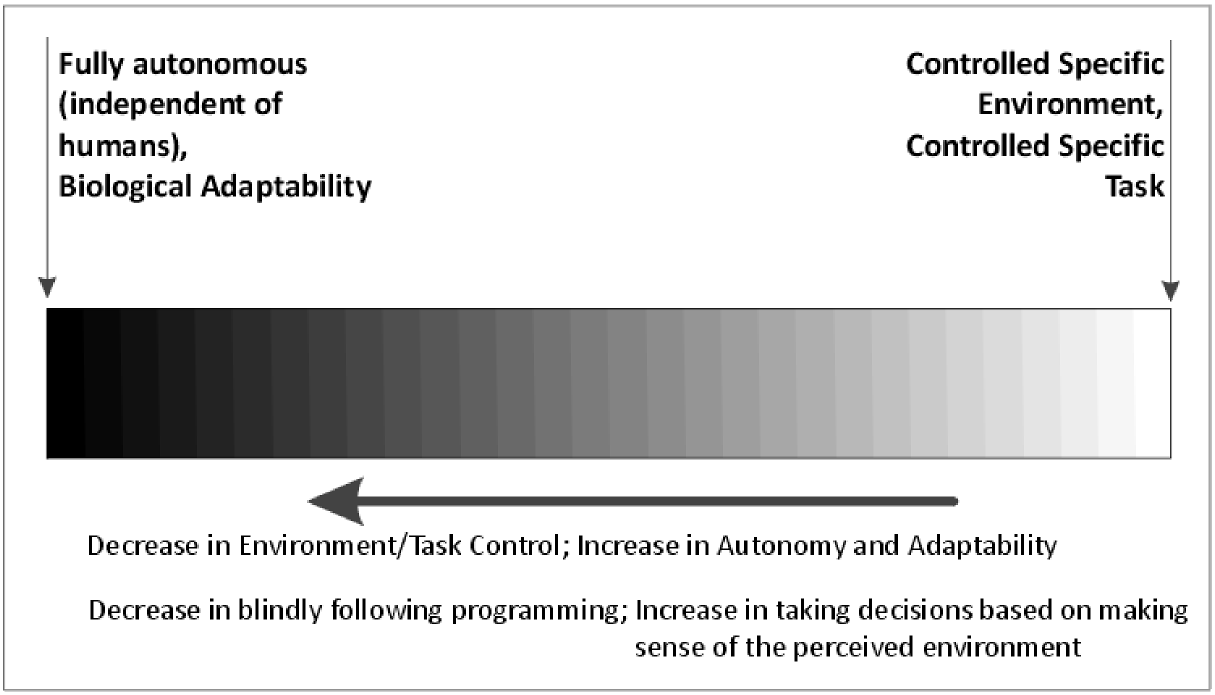


Figure 1: Control Spectrum of Behaviour

The Control Spectrum, as depicted in Figure 1, seems to suggest that, in order to get from the right to the left end of the spectrum, one simply has to walk along a path that is characterised by an incremental decrease in control, with every ceded control element replaced by an element of either autonomy or adaptability. This so far has been the point of view of those that base their designs on Industrial Robotics technology. A notable example is that of a recently published report by the European Robotics Technology Platform (EUROP) called “Robotic Visions to 2020 and beyond – The Strategic Research Agenda for robotics in Europe, 07/2009” [18].

The IMORGal theory, on the other hand, sees a different situation where the spectrum is divided into two sections. Each section represents a technological nature: to the right of the spectrum there is the Industrial Robotics technology with a nature of control; to the left there is the IMORGal technology or any other technology with a nature of the necessary biological principles, which the IMORGal theory summarises as autonomy (independence) and adaptability. This situation is shown in Figure 2.

The gap between these sections are called the Technological Gap. It speaks of a technological divide between two inherently incompatible natures, where one nature may try to yield solutions for the other nature but, in the end, find itself only realising a solution that is significantly inefficient and lacking in terms of performance. It is commonly known that what biology is good at, computers are not, and vice versa.

Figure 2 indicates how at the very right end of the spectrum we find all the vast number of Industrial Robotics products. As an attempt is made to move towards the left, ceding elements of control along the way, it becomes exponentially more difficult to design solutions that exhibit biological comparable behaviour. The IMORGal technology, on the other hand, starts with a base of autonomy and adaptability and, given that some control is still required for the initial simpler systems, the IMORGal technology starts off somewhere in the middle of the spectrum.

Note that the IMORGal technology is also subjected to an exponential decline – it is simply natural to expect that as the complexity of IMORGal solutions increases, the effort of realising those solutions will increase exponentially.

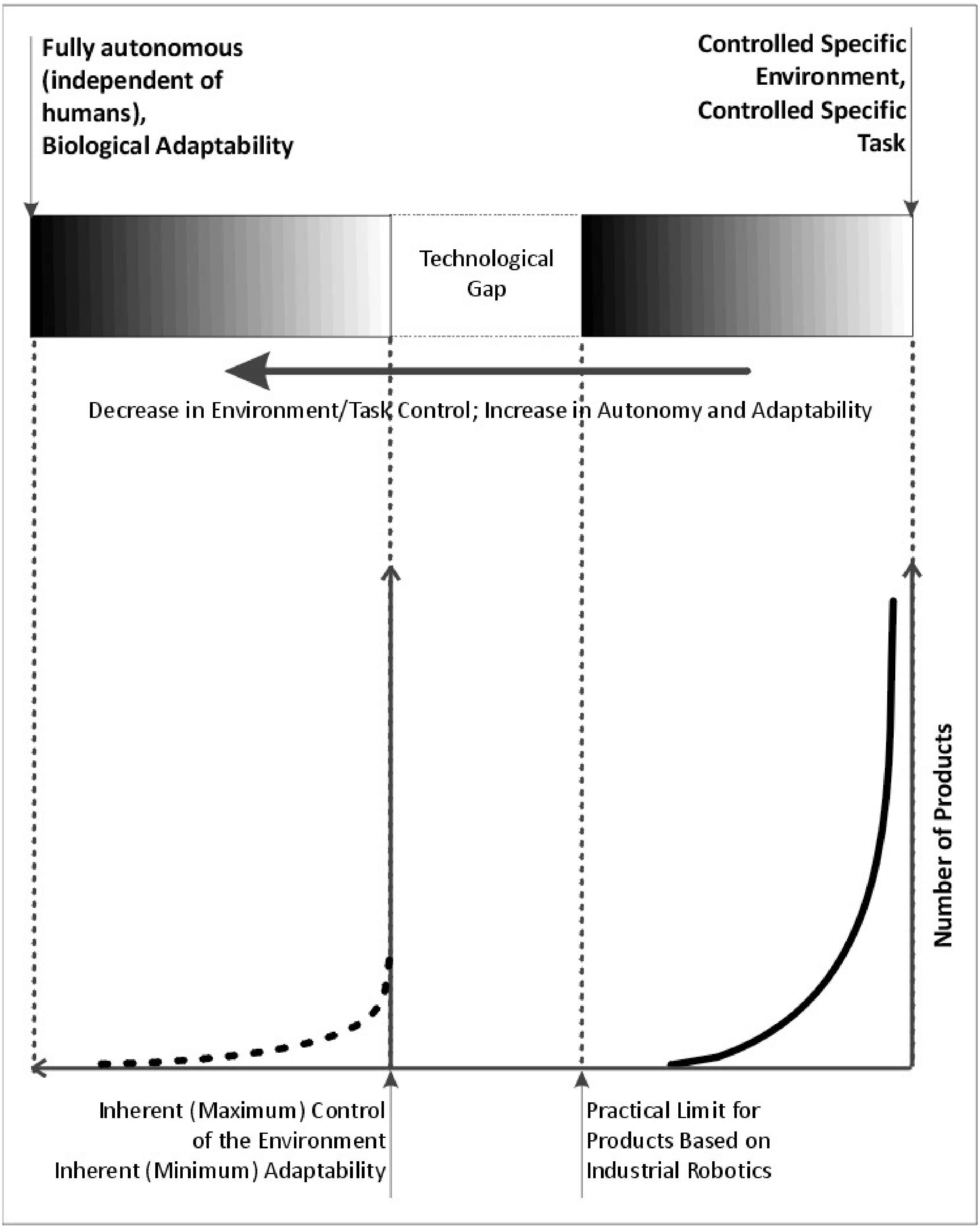


Figure 2: Control Spectrum of Behaviour according to the IMORGal Theory

In addition, note that the exponential curve is dotted: this conveys the predicted nature of the realised IMORGal solutions in that, since adaptability will be a fundamental part, a single solution can be used to solve various specific problems and be able to cope with differences in the environment of these problems. The solid line for Industrial Robotics technology refers to a myriad different technologies and products to solve for specific environments and tasks.

Therefore, the Industrial Robotics technology with its nature of control will not be considered for realising a technology that can reach biologically comparable behaviour. The IMORGal technology strives to provide an alternative solution.

2.4 Alternative Robotics Technologies

Do any other technologies or concepts exist that could be considered for reaching biologically comparable behaviour? Actually, there exists a plethora of different research efforts being done all over the world that step away from traditional concepts as found in Industrial Robotics. Are there any that have a real prospect of solving this behaviour problem, perhaps with some products already on the market? Not really (in a convincing way) because it appears, as shown above, that the research being converted into nonindustrial robotics products are mainly those based on Industrial Robotics technology. It seems to be a case of no one has an answer, so everyone has an answer.

Where there are some apparent potential, it seems that technology is limited still to a specific environment and task and/or heavy reliant on human support in order to function.

As I have said at the start, the IMORGal theory considers biology as a reference design and thus discards any behaviour-generating technology that fails to comply with that reference design. It is not good enough to simply incorporate a biological element or process (for example, evolution) – this element or process must also fulfil the same purpose in the larger picture of behaviour-generation than that which the element or process fulfils in the reference design.

A biological organism is not a bunch of components thrown together that you can rearrange differently to obtain a different organism. A biological organism is absolutely optimised and this implies that every single element or process has an exact optimised purposeful function. As with any other optimised design in electronics, the more you optimise, the less your design becomes flexible and the more difficult it becomes to apply line-for-line the elements and processes of that design to a different problem.

2.4.1. Evolutionary Robotics

Evolutionary Robotics is a relatively new field with impressive results. Harvey describes Evolutionary Robotics as follows during the interview on Talking Robots:

"Evolutionary robotics is an approach to designing robots whereby one tries to minimise the sort of preconceptions one puts in. So, naturally evolved animals and plants, there wasn't an external designer with a 'D', that planned how to put together the body and the muscles and the nervous system. Rather, evolution tinkered and played and put together whatever worked. And evolutionary robotics takes a similar stance, of saying "we don't really know how to put together a nervous system for a robot", rather than putting in preconceptions and designing them in by hand, let us merely set up the conditions where by artificial evolution has the possibility of evolving systems that work and then with minimal preconceptions set it going, press the go button, go away, let artificial evolution find what nervous system architectures and parameters it can. After the event check to what extent it works, perhaps analyse how it works (I have a slight prejudice often against analysing how it works), and it does free one from the typical GOFAI constraints of saying "if a robot is going to work then there is no alternative to making it work this way", where this way might include with internal representation or something."

NOTE: GOFAI refers to Good Old Fashioned Artificial Intelligence and is basically the traditional AI mentioned by Pfeifer above. The concept of representation or internal representation refers to a cornerstone element in traditional AI that relies on using an internal world model (internal to the robotic system) of the external environment for generating intelligence [10].

Another definition is [19]:

"Evolutionary Robotics is the application of artificial evolution to robotic systems with a sensory-motor interface to the world."

This field is vast and yields impressive work being done by people like Harvey and Floreano, with a large amount of literature available that includes using Evolutionary Robotics to study biology. One of the major challenges is that that some solutions leads to robots being optimised to a current specific environment at the end of the evolutionary process [19]. Changes in the environment (for instance, a different surface) could require that the evolutionary process be restarted. This process is not necessarily simple nor always of short duration. A new solution could require several generations with each generation requiring the individual to perform for a certain time such that its fitness can be evaluated. These issues are known and being addressed [19]. An additional issue is that of scalability (which is also being addressed). Solutions that might look promising in basic lab experiments, might not be that applicable to robots that must perform useful functions that require complex behaviour and sensorimotor processing.

In principle, the IMORGal theory differs from Evolutionary Robotics with regards to:

- The role of evolution during the realization of a robot and its behaviour
- The problem being solved, which ultimately dictates the starting point and the direction into which the corresponding technology is developed

Regarding the role of evolution: The issue is that Evolutionary Robotics starts with 'nothing'. You have no working/functioning/doing-something-useful individual to start with – it is via the application of an artificial evolutionary process that a functional individual is realised [19]. Note that this is specifically intended as such – the whole point is that of using artificial evolution to design the behaviour of the robot and its underlying mechanism. However, what I see in nature is that there are biological organisms and these already have a useful life, and evolution simply acts upon this individual. Regardless of how we came to be humans, one can safely say that at least during the last few millennia humans have not changed much genetically. We might be taller, a bit softer skinned, a slightly different hair type, weaker bone structure, and so forth, but we are architecturally or structurally the same: we still have five fingers on each hand, two eyes, we walk upright, the same internal organs, and so forth. All this time there has been no designing going on with respect to evolution designing a new species. Yet for many generations humans have been procreating, adapting and surviving – in all this time evolution was and continuous to be present but its role was not to design a new species (that is, if we speak in absolute terms and ignore for the moment any incremental designing by evolution that might exist).

The role of evolution was that of population adaptation, that is, evolution caused changes across a population such that the population as a whole could survive changes in the environment. While each individual organism has an enormous ability to adapt, it is effectively bound to the DNA from which it develops. Thus, to further increase the ability to adapt, that is, increase the probability of survival of the species, there exists evolution to perform this genetic adaptation such that genetically the population is adapting to its environment in a way not possible by the individual. Thus, the IMORGal theory sees evolution as of fundamental importance, and therefore evolutionary processes will feature in the IMORGal foundation, but as part of the bigger process of adaptation.

In fact, whether evolution is design or adaptation is irrelevant. The IMORGal theory says that, effectively, adaptation is a form of design. For example, let us say I design a robot for one environment, and the environment changes, so I design the robot for a second environment. If I designed the robot from the start to by itself adapt to the second environment, doesn't this mean that the adaptability I added was actually a case of me placing part of the design process inside the robot? That adaptability somehow equals design?

Additionally, the life span of the biological organism is what is important. For our purposes, the objective is that of building a useful nonbiological organism that during its lifetime can perform its function. However, if a biological organism dies, it dies and is gone forever from this earth – if a robotic system 'dies' whether by being switched off or running out of battery power, we can simply switch it on again and the robotic system is revived. If we are able to store the state of the robotic system's brain (all of its acquired experiences) before powering down, then it would not matter to focus on a single life span as with biological organisms. Additionally, when we resurrect the robot we can, at the cost of discarding the stored state, endow it with new DNA from our evolutionary process such that this new information will allow the robot to better function in its environment.

However, I would suspect that, in order to solve this 'useless individual' dilemma, Evolutionary Robotics might take a two-step approach (a possible example can be found in [19]) where the first part is sort of the design of the individual. During this phase, yes, the individual will not be useful at the start. However, it will be a useful individual when the evolutionary process ends, having acquired a set of innate behaviours or instincts. Thus, this individual can go out into the world and apply its usefulness and have a full life. The second part is sort of saying that during the life of the individual it could encounter changes that require additional adaptation (often referred to as ontogenetic learning) in order for the individual to continue being useful. This approach sounds promising but cannot be fully evaluated while applied only to simple robots and behaviours, as is the case of [20] where a robot has to navigate through a maze without touching the walls.

Regarding the problem being solved: Let us look at the range of research projects at the LIS directed by Floreano [21]. For example, there are:

- Division of Labour: *“Understanding division of labour in social insects using agent based models and artificial evolution”*
- Communication: *“Exploring the emergence and evolution of communication in social organisms using collective, evolutionary robotics”*
- Active Vision: *“The sequential and interactive process of selecting and analyzing parts of a visual scene”*
- Adaptation: *“A method to evolve the ability to "learn" on-line how to solve a task, instead of evolving a solution for a task as is common practice in evolutionary computation”*
- Learning: *“Evolution of Learning-Like Behaviours”*

Whereas several themes are researched under separate projects, the IMORGal theory says, firstly, that their exists a hierarchy with a single point or origin. From this origin all biological organism features sprout outwards, driven by necessity, and ever increasing in complexity. This is a difficult concept to explain. One argument goes as follows: I can build a useful individual without the ability to recognise other individuals of the same kind. However, if I now add this ability, I can expand the design of the individual such that the population as a whole can now have greater abilities than what is possible with an individual. By further adding communication skills I can continue to increase the design of the individual to once more improve the ability of the population.

Secondly, the IMORGal theory argues that all these different elements such as communication, cooperation, adaptation, learning and vision are present in every biological organism (note that the IMORGal theory sees learning as an advanced form of adaptation). Even more, all these elements are integrated in the biological organism: they are not modular units that can be developed separately and simply connected afterwards but there exists a balance between them that is specific, purposeful and required for successful functioning. Therefore, the IMORGal technology is about focussing on a single individual and expanding it until a complete nonbiological organism is yielded.

I must reiterate that it is currently difficult to clearly explain the concepts mentioned above. Only when the IMORGal foundation has been realised will it be possible to see via examples what I am trying to say here.

On the other hand, notice the power of Evolutionary Robotics (mentioned by Harvey above as well): instead of trying to design a circuit that determines how to process the inputs from the sensors into output values for the motors that will cause a robot to navigate through a maze and avoid contact with obstacles, an artificial neural network is set up and adjusted via the evolutionary algorithm to yield a design that by nature is optimised. Just imagine how powerful the result will be if we start with an already complex and functional almost-complete population of individuals (having used explicit design to solve the easier parameters that need to be optimised) and then use the evolutionary process to optimise/design for us the more difficult parameters. The easier parameters could be those required immediately by the robot to have a useful life. The more difficult parameters could be those that affect more the structure of the robot's design (even hardware: number of sensors, types of sensors, types of motors, size and weight of the robot). Ideally we would want to take into account the rate of change of the environmental factors to which the robot has to adapt: the factors that change rapidly (from milliseconds to days) should be dealt with by the individual through adaptation of the easier parameters; the factors that change across several generations should be dealt with by the population through adaptation of the difficult parameters. For situations like these where one considers how to apply adaptation, it helps to think of a case where the robots are forced to be independent from humans. Space exploration is the best example. Imagine sending a spacecraft filled with robots to a planet in another solar system. The objective of this robotics community is to build a station inhabitable by humans within 50 years, at which time the first humans are expected to arrive. This is basically the target situation which the IMORGal technology intends to ultimately solve for.

2.4.2. Behaviour-Based Robotics

I have mentioned Behaviour-Based Robotics above and the importance it places on embodiment and a close link between perception and action. An additional element is that of minimal sensing, that is, to implement only the minimum level of processing necessary for the robotic system to perform its function [10].

The IMORGal theory agrees with these concepts. As mentioned before, the IMORGal theory goes even further to say that, for the sake of yielding an efficient solution, every bit of information that can be extracted from the environment with the given sensors, should be extracted by the designed nervous system of the robot. Thus, by choosing a sensor, you are choosing by default a perception set.

Speaking about nervous systems, there is a fundamental issue regarding Behaviour-Based Robotics' implementation of the nervous system for the robot. In Behaviour-Based Robotics there is no centralised brain – what is preferred is a distributed processing architecture [22]. It is well known that free-moving biological organisms all have a centralised brain [23], although a small part of the nervous system also extends down the length of the organism (the spine, in the case of humans). If biology is so extremely optimised, than this centralised architecture is not just some solution, it is the best solution and not one of several possibilities that nature just happened to choose. It is centralised for a reason. Therefore, the IMORGal theory only considers a centralised nervous system as an architecture that complies with the biological framework.

An additional issue mentioned in the literature [24] is that Behaviour-Based Robotics has limited adaptability because there is no learning from experience. Learning is such an important concept in especially humans that leaving it out is not an option. However, learning is based on memory, and memory comes in different types. The memory effect can be defined as the situation where an event in the past affects behaviour in the present. The IMORGal theory says that the least complex memory effect should be implemented first. Following an incremental approach, the most complex memory effect (as most likely found with humans) should be implemented last. The idea is that if the least complex memory cannot be incorporated successfully, then any realisation of more complex memory will be inefficient and the design overall incomplete. It is suspected that all memory types, from the least to the most complex, exist in the human brain.

On the other hand, Behaviour-Based Robotics is the most practical field regarding an attempt at generating biologically comparable behaviour and speaks for itself when one looks at the success of the iRobot products. I agree in general with most of the principles upheld by Brooks but, unfortunately, due to time and space limitations I will not discuss this technology in detail here.

Scalability is a known problem with Behaviour-Based Robotics [33] and since scalability is fundamentally important to the IMORGal technology (and taking into account other issues mentioned above), Behaviour-Based Robotics is seen as a reference instead of an applicable technology.

2.4.3. Neurorobotics

The last robotics technology I want to discuss is that of Neurorobotics [25], a field in which Gerald Edelman's group at the Neurosciences Institute in La Jolla, California [26], has and continues to play a pioneering role. Neuroscience, in particular Neurobiology, sits at the heart of Neurorobotics; accordingly, the designed nervous system (being the main focus) is based on neurobiological principles. The developed neurobiological model of a particular biological nervous system is then combined with a physical (that is, embodied) robotic mobile platform with sufficient sensors and actuators to conduct experiments. These robots are referred to as neurorobotic devices in general. Brain-Based Devices (BBD) are a class of neurorobotics which emphasises interaction with the physical environment and having a basis of anatomical and physiological features of vertebrate nervous systems [27]. BBDs are constrained by the following design principals:

- *The device needs to be situated in a physical environment.*
- *The device needs to engage in a behavioural task.*
- *The device's behaviour must be controlled by a simulated nervous system having a design that reflects the brain's architecture and dynamics.*
- *The behaviour of the device and the activity of its simulated nervous system must allow comparisons with empirical data.*

An example of a BBD is Darwin X [27] which was developed by Edelman's group at the Neurosciences Institute. Darwin X was used, amongst other purposes, for investigating the role of the hippocampus in the formation of episodic memories. Episodic memories are what allow us to relate a specific event in the past to time and space, that is, our sense of remembering when and where something happened with respect to our present time and location or sequences of locations. The simulated brain of Darwin X contained 50 distinctive neural areas, 90.000 neuronal units, approximately 1.4 million synaptic connections and was modeled after the anatomy and physiology of the mammalian nervous system but with far fewer neurons and a much less complex architecture. Each neural area contains neuronal units that can be either excitatory or inhibitory. 'Simulated' means the brain model was computerised and executed not on the robotic mobile base but on an external computer cluster that interfaced with the physical robot via wireless communication. The external computer cluster (a Beowulf cluster containing 12 1.4-GHz Pentium IV computers) was necessary since the model is computationally intensive but, more importantly, allowed the possibility to record every event that took place inside the simulated brain for the purposes of post-simulation evaluation. The corresponding computer power allowed real time interaction between the robot and the simulation but with a simulation cycle of 200 ms. During each cycle the inputs are read, internal model states updated and output generated for the wheels. Physically Darwin X is basically a typical trash-can shaped robot with a wheeled mobile base. In terms of sensors it has a CCD camera, an odometer, and infrared transceivers for object detection. Thus, a wonderful setup for neurobiological research.

How does Neurorobotics compare to the IMORGal technology? Firstly, there is agreement that the robotic system must be embodied and have a nervous system based on Neurobiology. However, it is important to note that Neurorobotics is basically a research field. Nevertheless, there is an expectation for incorporating BBD principles in a conventional engineering system to the extent that the engineering system can benefit from concepts such as learning by conditioning and long-term memory combined with episodic memory [28]. The concept of a hybrid is normally mentioned. Thus, the first issue with Neurorobotics is that it is too experiment based, too focused on toy real worlds as mentioned by Holland above. The problem is that a real-world complete robotics product is a significantly different challenge than a laboratory experiment. Even though the hybrid concept is about only applying the Neurorobotics principles instead of using a specific neurorobot constructed for the purposes of an experiment, the mere fact of lacking a complete robotics product leaves the door wide open for an inefficient technology as a result. The specific higher brain function being modelled is treated too much in isolation from the rest of the brain as though it is possible to extract only that part, model it, implement it and yield a system that now contains that brain function in a useful way.

Neurorobotics does not start small but immediately attempts to model higher brain functions of mostly the human brain. Of all biological brains, knowledge of the human brain is most valuable to us since, for one, it is the most intelligent brain by far. However, it is also the most complex. The IMORGal theory says that it is imperative to start at the smallest and least complex scale possible. The question then is: Why can't we start at the top? Why do we have to start at the bottom? Note that the human brain is not a unique brain that is different from other biological brains. The human brain shares similarities with other mammals, from rodents to primates. The human brain has areas that correspond to the reptilian brain as well. It appears, architecturally speaking, that the human brain is merely an extension of the brains of animals with inferior complexity. This of course makes sense from an evolutionary point of view. Movements for example can be controlled by subcortical areas but in order to achieve the precise movements possible with a human hand, these movements are overridden by activity in the human brain's prefrontal cortex. Here are two arguments why we should not start at the higher brain functions:

- Modularity. The brain might have many regions but it is not modular. That is, there are no clearly identifiable interfaces. It is one big entangled yet organised mesh (in spite of the neatly organised 6-layer cortex with its cortical columns). Since the interfaces are as such, any attempt to force the brain regions to be modular inherently incurs a significant penalty. So much so that the synthesised implementation of the modularised region yields performance far inferior to the biological reference. In addition, you end up with an implementation that lacks the basic sense of its purpose in the global picture of the organism, that is, the sense that we are familiar with when observing the reference biological organism. For example, memory and its sense and how the hippocampus and cortex depends on the nature of information flowing into it and how this information is of a specific type that relates back to the specific organism and its morphology which relates back to the organism's purpose.

- Context. The concept of sense above flows into the more palpable concept of context. Context says that the local characteristics and meaning of signals that enter and leave the region is different from the global characteristics and meaning of the organism as a whole, that is, looking globally at the purpose of that region and its interaction with the rest of the brain. It is as though the purpose of the region is defined or a natural extension of the source regions. Different source regions would change the region in question in some way.

Many try to model the mammalian cortex due to its homogeneous structure and being the storage area for our memories. If we were to have an anatomically and physiologically correct model of the human cortex including the eyes and the complete visual processing system, does that make it possible to add human-level visual and memory abilities to a robotic system? Have we considered some of the more subtle aspects related to vision and memory? For example:

- We never process a complete image, like a camera image. Our eyes always focus on only a single point in the image. We can focus on a small specific detail at close range, or we can look out over the landscape and at a point into the distance. Sometimes we can focus our 'eyes' on a side part of the vision: our eyes focus on an object directly in front, but we are actually looking at something to the side, something we cannot clearly see because of the physical image distortion of the eye image. Sometimes we don't see anything outside in the real world because we are looking inside our memory, for example, remembering the face of somebody. At that exact moment of seeing that face in front of us, we cannot see a real world object. When considering all these variations, which part of the brain is deciding what to focus on?
- Head movement, or any other movement that we make to see better, squinting of the eyes, staring in a good (intensely studying an object or perhaps simply reading) or bad (at an attractive woman) sense. What links the visual processing to these motor functions?
- We don't remember everything we see. Sometimes we wish we could stop remembering an unpleasant event like seeing a car accident, or the face of someone we disappointed or hurt. We can willfully try to remember something, like someone's phone number written on a piece of paper, or looking at a presentation, or a teacher in front of a class. We pass through situations where someone sees and notices something, a situation or some dodgy character passing us by, or something beautiful. What determines the extent to which the brain is going to remember something?
- Why do we remember one thing but not another? Why does a certain object or situation have importance above another? How does the meaning or symbolic value of an object affect the extent to which I remember that object? Or simply that which catches my attention? That which I decide to focus my attention on? What decides this? What determines this? Why do I even care what I remember?

What other small indirect details about higher brain functions are required to have a complete model? More importantly, what controls these details? Isn't it perhaps controlled (or their actual effect determined) by lower more primitive areas in the brain? There is this seemingly intricate interaction inside the brain, a relatedness between higher and lower brain areas. It is as though the higher brain areas are in service of the lower brain areas, they provide the more advanced perception that allows us to extend the complexity of our intelligence way beyond that of animals. How can we thus have a complete model of the higher brain without having a complete and accurate model of the lower brain areas?

Jeff Krichmar, from the University of California, Irvine and previously part of the team at the Neurosciences Institute, is a key contributor to the field of neurorobotics and since recently focuses on the role of neuromodulators inside the vertebrate brain [24]. The neuromodulatory systems for vertebrates are found in the sub-cortical areas (that is, the 'lower' brain structure in the sense of being below the cortex, roughly speaking). The neuromodulatory systems play a fundamental role in regulating behaviour and influencing decision-making. Krichmar then states as follows:

“Moreover, the neuromodulatory systems provide the <u>foundation</u> for cognitive function in higher organisms. Attention, emotion, goal-directed behavior, and decision making all derive from the <u>interaction</u> between the neuromodulatory systems and areas such as the amygdala, frontal cortex, and hippocampus. Therefore, understanding neuromodulatory function may provide a <u>basis</u> for the construction of cognitive machines and the control of autonomous systems.”

In terms of the IMORGal theory, the important role of the neuromodulatory systems confirms that starting with higher brain functions will lead to an inefficient technology. Then, the IMORGal theory goes further:

- Neurorobotics is mostly about models of specific areas in the brain. However, is it certain that we will eventually need an exact model of such areas when we want to incorporate the corresponding functions into a robotic system? Most agree that we do not want to copy biology; we don't want to rebuild a specific biological organism using biological means. We want something else, a different creature with a different purpose and usefulness to us but still able to exhibit biologically comparable behaviour. Given the vast difference in morphology to be expected between a robotic system and biological organisms, it is likely, should we want to implement for instance episodic memory, that an exact neuronal model of the hippocampus is an inefficient solution.
- Every element of the human brain has its purpose. There is a reason why our memory system is different from that of other mammals (to a slight extent) and insects or reptiles (to a significant extent). The IMORGal theory says that the type of memory and its architecture depends on the level of autonomy and adaptability that we want. As the IMORGal technology increases in complexity, the need for more complex sensor processing, motor control and adaptation will lead to a blossoming of the previous level of IMORGal complexity and projects outwards.

Do we really need a vision system that has a CCD camera and in the order of thousands of neurons to add long-term memory? The challenge for practical robots is to try and find the minimum elements necessary to implement such a function. Thus, the downward scalability of Neurorobotics is an issue for the IMORGal technology.

2.4.4. Action Selection

In spite of all the comments made so far about any technology, one specific issue has not been dealt with yet – this issue is arguable the most important of all. It simply has to do with the reason for doing anything. It is about answering the question “Why?”. Why is a robotic system performing a certain action at a specific point in time? What drives that action? What is the process inside the robot's nervous system or controller that causes this action? BBDs are said to be fundamentally different from conventional AI robots because its controller is not a sequence of programmed instructions [28]. The 'controller' for Darwin X, for example, includes behaviours such as exploration and obstacle detection and avoidance. Its default behavioural sequence is as follows:

- Move forward for 10s
- Rotate 60° to the left and wait for 3s
- Rotate 60° to the right and wait for 3s. Repeat.
- Rotate 60° to the left and wait for 3s
- Calculate a new heading based on objects detected during the rotations. Select:
  - (1) same heading,
  - (2) 60° to the left of previous heading or
  - (3) 60° to the right of previous heading

What drives these actions? The nervous system of Darwin X contains an action-selection system. It turns out that Action Selection is a field of research that is directly relevant to Neurorobotics [29]. Action Selection can be informally described as [30]:

“The task of choosing 'what to do next'”.

A more formal definition is:

“Given an agent with a repertoire of available actions, some knowledge of its internal state, and some sensory information concerning environmental context, the task is to decide what action (or action sequence) to perform in order for that agent to best achieve its goals.”

An 'agent' for the purposes of this document refers to the general concept of a robot.

Thus, the field of Action Selection does appear to be concerned with what drives action. However, the specific word use hints towards a more programmatic/algorithmic/logical approach that is more concerned with answering “How?” than “Why?”. Too much emphasis is placed on the solution being a control system – 'control' is the fundamentally element that the IMORGal theory tries to get away from. It appears the hierarchical breakdown of actions into base units is not clear (for example, what motor level commands and sequence constitute the action of taking a drink from a cup?). No satisfactory answer to the question of “Why?” was found in this field – the IMORGal theory proposes a solution in the following section.

2.5 The IMORGal Theory Speaks

2.5.1. What is the problem being solved?

Let us start with a general question that to an extent voices the general opinion concerning machines and biology. I will thereafter rephrase the question several times, attempting to highlight how the IMORGal theory interpret the problem and what approach the IMORGal theory follows in order to provide a solution.

(Q1) What elements of our understanding of biology can help us build robots that can function as well as biology in dynamic uncertain environments?

(Q2) What elements of our understanding of biological organisms can help us build nonbiological organisms that can function as well as biology in dynamic uncertain environments?

From Q1 to Q2 we see that the IMORGal theory focuses on the biological organism level, that is, the biological world in which organisms interact with other organisms and their environment. The robot is seen as simply one more organism in the biological world. Therefore, the emphasis is on the robot to behave like biological organisms in order to blend in with the biological world. Instead of us becoming more like robots, it is robots that have to become more like us. The basic concept of an organism is simply that of an entity that complies to the following model:

- External environment ->
- Physical boundary (body) ->
- Internal environment.

Via the boundary, perception of the external environment enters into the organism and, using physical means, the organism exerts itself onto the external environment (allowing the organism to possibly change the external environment). This model is true for biological organisms from one-celled (bacteria) to human.

(Q3) What elements of our understanding of free-moving biological organisms can help us develop synthetic organisms that can function as well as biology in dynamic uncertain environments?

From Q2 to Q3 we see that the IMORGal theory narrows the organism focus even further. The interest is not any biological organism but specifically those that are free-moving, for instance, including humans, animals, insects and bacteria but excluding plants. What this means is that emphasis is placed on the biological organism's nervous system. All free-moving biological organisms have effectively a centralised nervous system [23], from bacteria to humans.

Thus, the basic functional model applied contains simply the following elements:

- Sensors, for perceiving the external as well as the internal environment
- Actuators, for the organism to exert itself onto the external as well as the internal environment
- Nervous System (or Brain)

The IMORGal theory says there must be a reason why free-moving biological organisms have an effectively centralised nervous system. It is as though the nervous system has to do with some form of information processing and the optimal architecture required is to have a single point of interconnection and correlation between sensors and actuators to such an extent that sensing and action appear to happen simultaneously.

However, the IMORGal theory does not limit itself to constructing only free-moving nonbiological organisms – included are systems that have an actuation system that physically effect the external environment, for instance, an automatic revolving door at a building entrance (the door has sensors to perceive the presence of people, a 'brain' to control the actuation system which is to turn the physical door, which is a physical effect on the real world allowing people to enter or exit the building). Note that the concept of a 'body' does not exist as what is typical for biological organism. Yet, a 'body' is a word we humans have invented to signify the physical outer boundary of a biological organism. This boundary is imperative in the biological world, related to the nature of biology, but does not have to be a limitation for nonbiological organisms.

Since, the term 'robot' can be misleading sometimes or open to biased interpretation, the IMORGal theory prefers the term 'synthetic organism' to emphasise the idea of building a physical entity that complies to the general concept of an organism.

(Q4) What are the necessary fundamentals of biology's reference design for free-moving biological organisms that are required to develop synthetic organisms that have biologically comparable behaviour?

As mentioned previously, biology is seen as a super-optimised solution and any attempt to develop significantly complex synthetic organisms that are not fundamentally based on biology, will ultimately lead to inefficient solutions. Thus, free-moving biological organisms form a reference design. The emphasis is here on biology being a 'reference' as opposed to being instructions for developing synthetic organisms. The idea is not to copy biology, to reproduce biology, to build more humans or more animals. The morphology and ontogeny (that is, the development of an individual organism from embryo to adult) are vastly different for biological versus nonbiological organisms. Note the following:

- The synthetic organism starts off as an adult instead of being born from a cell.
- Biological organisms are forced to reproduce by themselves. A synthetic organism is born on a manufacturing line.
- The complete biological organism is 'obsessed' with survival. The purpose of a synthetic organism is different – somehow it must do something useful to us the consumer.
- The synthetic organism will most likely be constructed from nonbiological materials.

Thus, it is the fundamentals that are important, not the physical realisation specifics *per se*. Additionally, of all the biological organism fundamentals, it is only their behaviour that the IMORGal theory is interested in – these are the necessary fundamentals.

(Q5) What are the necessary fundamentals of biology's reference design for free-moving biological organisms that are required to develop practical and useful synthetic organisms for the real world that have biologically comparable behaviour?

From Q4 to Q5, the IMORGal theory emphasises the need for a technology that will allow the development of products that are practical and useful to us in the real world. Given the expected significant complexity that these organisms could have, it is inherently required that the solution, especially the brain design, be extremely efficient in terms of yielding the desired complex behaviour without requiring by default a large and power-hungry system. If someone thinks adding so much complexity to a simple revolving door is insane, note the following: the current controller device (for example, a PLC) for each of the latest Metro/Underground train doors, are probably more powerful computationally speaking than the bulky room-sized mainframes of several decades ago.

In terms of useful products, the IMORGal technology aims at offering solutions where the distinct feature is that of a system that adapts to its environment instead of the environment (which includes people) adapting to the system. This adaptability will go hand in hand with the concept of 'making sense'. Rather than simply following a state machine approach where the next state (which will determine the next action, if any) is updated from the current state according to designed conditions, the IMORGal approach is that at any time the synthetic organism will make sense of what is going on around it at that moment, taking into account its internal state, and take appropriate action. The inherent problem with a state machine is that if the system starts getting complex, the number of states can be significantly large, to such an extent that testing for every single possible state is impractical. In reality it could and does happen that a machine sometimes gets 'stuck' inside some loop because of some state not tested for.

Where there are systems that only require a part to have biologically comparable behaviour, for instance the HMI of electronic equipment, it can be considered to apply the IMORGal technological concepts to the HMI and allow for an interface between the HMI and the rest of the equipment.

The IMORGal technology firstly produces an abstract model, that is, a solution not bounded to a specific physical realisation technology, thereby allowing the same model to be ported to a suitable physical medium. The second part involves realising the model in the physical medium. The details of this process will be shown in the synthesis section.

2.5.2. The Need Theory

From the previous section we see that solving for biologically comparable behaviour is essentially a brain design problem. How should one proceed from here? The problem that we face is colossal: on the one hand we have the biological brain (not just one but a vast amount of brains which range from being slightly to almost completely different from one another), and on the other hand we have the yet undefined architecture of the synthetic brain. What could possibly be the starting point for designing a synthetic brain that uses a biological brain as reference? What will be the origin from which the the brain design sprouts outward to incorporate greater ability?

Should we start with neurons? The IMORGal theory says that neurons are too physical-medium specific, too closely coupled to biology and thus starting off by forming different neuronal structures is prone to inefficiency. The main issue is not a lack in knowledge about a single neuron – the neuron has already been studied in the greatest of detail. If the key to unlocking biological intelligence was held within the structure and operation of a single neuron (including its pre- and postsynaptic connections), then we would have discovered it a long time ago. What is lacking is a concept that is fundamental but is not described in terms of neurons. Yet the concept must be so fundamental that it results in being as low level as that of neurons in the brain.

Let us consider the following situation: there is a robot (let us call it a robot here for practical purposes) located in the middle of some room. It has its body, and it has its sensors and actuators connected to its currently-empty synthetic brain. There is no computer program to instruct the robot to perform an action because the whole idea is to get away from control as far as possible. How do we get that robot to do anything? How can it generate action by itself? What is the driving element behind the action? If behaviour is emergent, what process can we place inside its brain that will have the effect of action?

I switch on the robot. What is the first thing it does and what makes it initiate that specific action without a programmed software instruction?

Let us consider a specific robot, one that I will use during the remainder of the document as my example IMORGal. It is called Roamer because its purpose in life is to roam around the space it finds itself in. It wants to always move forward and never stand still unless there is a reason to do so. Looking from above, it has a round shape (Figure 3), with a single infrared distance sensor located in front at the centre. The only requirement for the sensor is that it must provide an output value that is unsigned and ranges from 0 to MAX_DISTANCE, where 0 means the object is right in front of the sensor, almost touching the sensor. As the distance measured to the object increases, the value measured by the sensor increases exponentially until reaching MAX_DISTANCE which is the maximum distance that the sensor can measure.

Regarding actuators, Roamer has two wheels for driving which are located on its left and right side. Each wheel is driven by a separate simple DC motor whose speed is determined by a Pulse-Width Modulated (PWM) signal. By reversing the polarity applied to the DC motor, the wheel turning direction can be set as either forward or reverse.

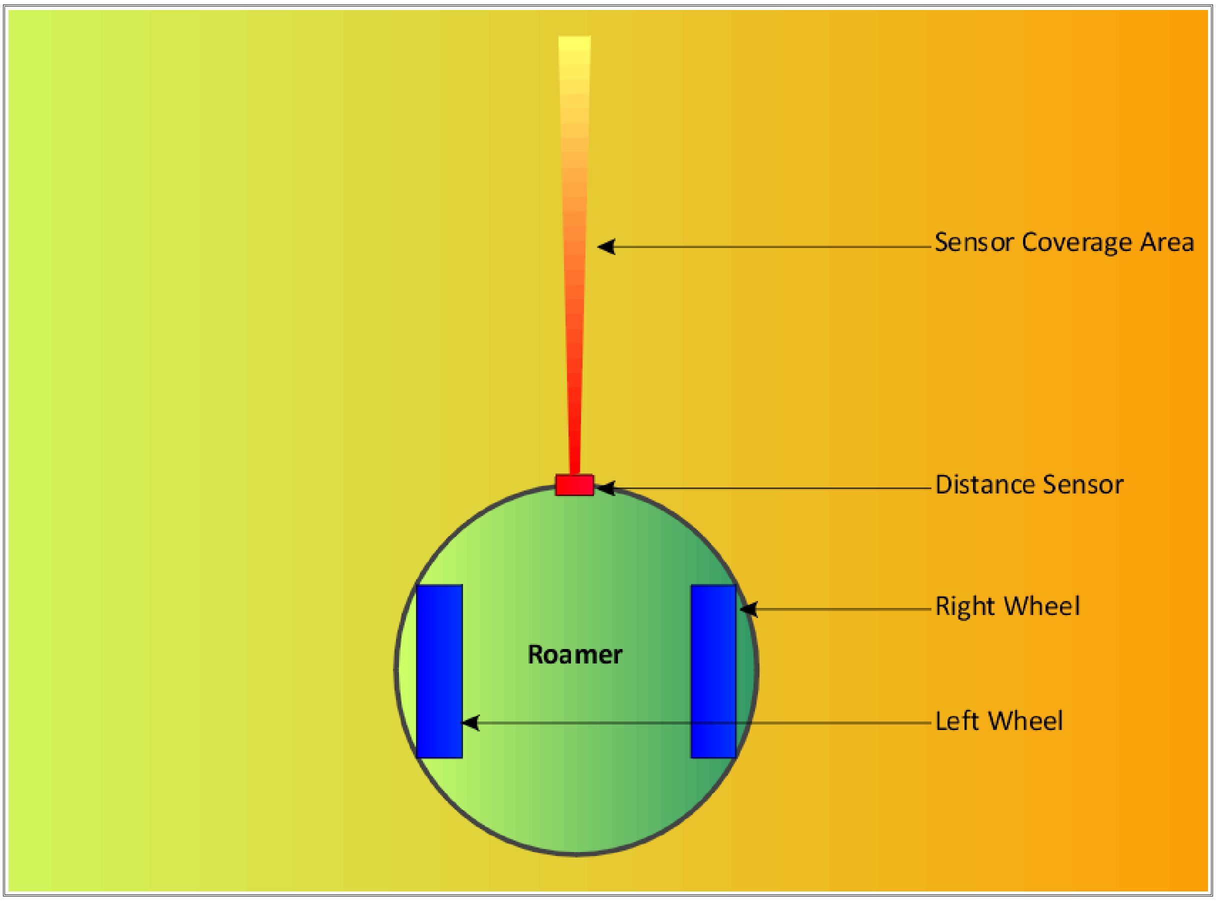


Figure 3: Roamer

Since each DC motor has a unique value associated with it that is independent from the value of the other, Roamer effectively has a differential drive which allows for turning. A small driveless wheel is located at the front and rear (not shown in the figure) for balancing purposes. Thus, Roamer is as simple as can be, containing one sensor and, effectively, one actuator.

Returning to the matter of finding a way for Roamer to move, I will use a dog as an example to help us understand the concept at hand. If there was a dog in the middle of the room, what action would it be performing next and why would it initiate that action? Imagine the dog was asleep and has just woken up. Let us imagine its first action was to get up and walk over to the water bowl and drink some water. We might imagine the reason was obvious: it was thirsty. However, let us look at the situation in the following way: during the time while the dog was waking up, its bodily processes were busy functioning and these detected that its body was lacking water. At this point, some alarm signal appears in some way inside the dog's brain to such an extent that the dog perceives this signal, recognises it means it is thirsty and knows what to do to stop the alarm signal. Notice that the level of water inside the dog is constantly changing in a periodic manner: as soon as the level falls below a certain threshold, the system inside the brain monitoring the level of water sets off the alarm signal to communicate this situation to the rest of the brain. The dog reacts and drinks water which causes the level of water inside its body to start rising again until reaching a value that satisfies what is required by the water monitoring system to accordingly cease the alarm signal. Now, as time passes that value of the water level starts to rise slower and slower until reaching a climax after which the value starts falling again, heading towards the threshold that will trigger once again the action of the dog drinking water.

One might think the situation is reaction-based but one can also look at it as need-based. That is, the dog has a need for maintaining a sufficient level of water in its body. It needs this in order to survive. One can view this water need to be in harmony – satisfied, in balance – when there is enough water in its body. As soon as that value becomes too low, the need falls out of harmony – becomes unsatisfied, out of balance – upon which it prompts the rest of the brain (via the alarm signal) to take some sort of action that would cause the necessary input (the water drank by the dog) to bring it (the need) back into balance.

Notice that the action to 'drink water' requires a significant amount of coordinated motor functions so that the correct muscles are excited at the correct time and in the correct sequence such that the dog can get up, walk over to the water bowl, bend down and drink the water. All these sub-actions are ongoing while the need maintains its alarm signal asserted.

That is the basic description of what I call a Need. The Need represents some value in the brain that constantly tends to fall out of balance if the required input is not received. Once out of balance, the Need transmits an output signal to the rest of the brain, prompting the brain to perform an action that would cause the required input so that the Need's value can rise to a value where the Need is in balance again and thus ceases asserting (it deasserts) its output. Viewed in a certain way, it is not an input-output model but an output-input model.

I suspect that every action taken by any biological organism is done in order to bring some Need back into balance. You can apply it to every single element of a human being: sleeping, eating, scratching one's head, blinking, breathing, laughing, falling in love, and so forth. Behind every action you will be able to think of a corresponding Need that is currently out of balance. It just so happens that in order to satisfy certain Needs, a potentially large set of coordinated actions might be required. Compare birds that fly to the other hemisphere to avoid winter or an animal mating ritual.

In fact, the Need theory can be used to specifically model the physiology of water retention in the human body. We have a hormone called Vasopressin (also called Arginine Vasopressin or Antidiuretic Hormone (ADH)) [34]. It is released into the bloodstream by the posterior pituitary gland in the brain, with the amount being released at any point in time controlled by another brain region called the hypothalamus. The hypothalamus is a small area at the base of the brain that controls functions such as water balance, feeding, temperature, our biological clock and stress responses. Roughly speaking, if the water level in our body drops below a threshold, the hypothalamus, who continuously measures the water level state, will cause the posterior pituitary gland to increase the amount of Vasopressin released into the bloodstream, which in turn will enable more water being retained by the kidneys and thus less urine to be generated. Of course, this just retains as much as possible the water already in the body. It could be that if the water level keeps falling, we need to perform some action to recuperate the water loss. Thus, we start feeling thirsty. Therefore there is a dual mechanism happening inside the brain: the water level is controlled with respect to the internal environment (Vasopressin) as well as the external environment (by drinking water). However, note that the same Need theory applies to (drives) both these cases. That in itself makes a lot of sense: why would the driving force behind behaviours that act on the internal environment be different from behaviours that act on the external environment?

It appears there is some sort of hierarchy in terms of certain Needs being more important than others, such that the action performed by the brain strives to satisfy those of higher importance first. For instance, if while on its way to the water bowl the dog heard a sound behind it which it didn't know, I suspect the dog would turn around and immediately become extremely alert, suspecting some sort of danger. The unknown sound does potentially pose a greater risk to its survival than its lack of water. This hierarchical nature allows me to argue that there are possibly many Needs inside the brain, not all positioned along a flat structure.

Therefore, this situation of a brain constantly striving to keep all its Needs in balance, forms the foundation for generating behaviour in the synthetic and biological organism. Due to its importance, the synthetic organism is called an IMbalance ORGanism which leads to the name IMORGal implying a synthetic organism designed on the principles of the IMORGal theory.

Thus, back with Roamer, how can I create a Need for it that will cause it to continuously move forward? This is why it is useful for an IMORGal to have a purpose in life because else it would not really need to do anything. Roamer needs to move forward. So, let us create a Forward Movement (FM) Need for Roamer of which the Value of the Need increases when there is a change in the sensor's output (either Roamer or the object is moving), and decreases if the sensor's output remains constant (Roamer is not moving). When the FM Need's Value falls below a certain threshold, it asserts its output (also called firing). This output is the prompt for the rest of the brain to cause the wheels to rotate so that the sensor's output will change.

What we would expect to see is the following (assuming there is an unmovable object in front of Roamer within its infinitely long sensor coverage area, indicated in Figure 3: we power up Roamer, it stands still for a moment, suddenly it moves forward closer to the object and stops after a while, it stands still again, it suddenly moves forward again, it stops, it moves forward again, and so forth. Should Roamer be able to reach the object in front of it, one would expect Roamer to not be able to move forward again as any forward movement is blocked by the object and thus causes no difference in the value measured by the sensor. What would happen is that the Need will never be satisfied again until Roamer's battery runs flat.

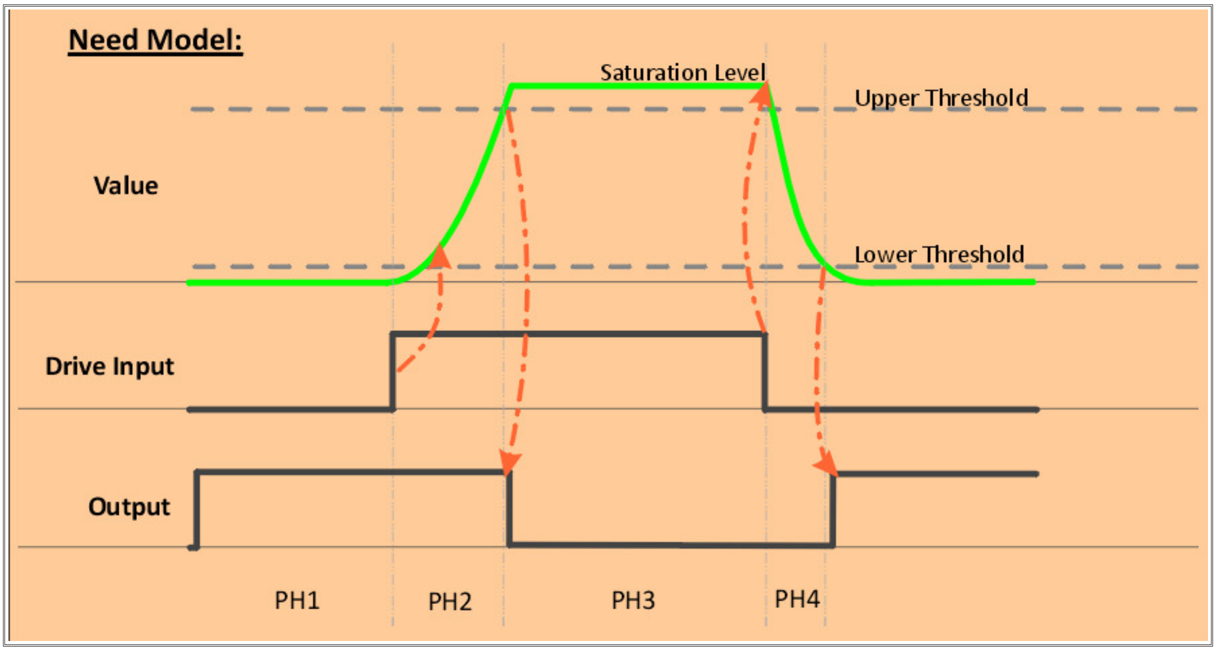


Figure 4: The Need Model

Figure 4 shows a plot of the Need model against time, including the Value and input and output signals. The basic cycle consists of 4 phases (PH1 to PH4). The Value increases when there is a nonzero input signal and decreases otherwise. The Value has a physical limit as to the amount it can reach; if the input signal is present (asserted) long enough, the Value reaches a maximum value which is maintained while the input is asserted.

In a state of imbalance, the Need output is asserted. That is the method of communicating this state to the rest of the brain. The concept of hysteresis is used to determine the changing between states of balance and imbalance.

While in a state of imbalance, the Need changes to a state of balance when the Value reaches the upper threshold, and thus the output is deasserted (set to zero). In a state of balance, the Need changes to a state of imbalance when the Value reaches the lower threshold, and thus the output is asserted.

Therefore, we now have a synthetic organism that is able to initiate purposeful (albeit not very useful) action based on the Need model rather than executing any software. It is the simplest IMORGal possible: one sensor, one actuator and one Need. Notice that the Need is described in biologically-compatible terms: values that rise and fall, and hysteresis. We know that a form of hysteresis exists with neurons. An example is the Threshold Stimulus which is the minimum stimulus needed to achieve an action potential (the causes the neuron to fire). In the Need model there is no mention of any binary values, logic functions nor any algorithm that specifies a computational method required for implementing the Need. The input and output signals do not need to be constant and can have arbitrary values.

At this point, although treated in more detail further below, I would like to mention the IMORGal's external environment and how I define it. The external environment is viewed as such that it is defined by the Needs of the IMORGal. The current Roamer is 'designed' for an environment that has a flat nonslip surface, of infinite size and has an object in front of it which falls within the sensor coverage area yet which somehow never allows Roamer to reach it before Roamer's battery runs flat. Needless to say, this is a description of an impractical environment and is a direct reflection of the lack in behaviour complexity of the current Roamer. The next Roamer version, the first real Roamer effectively, is a first serious attempt at a more practical and, therefore, more complex IMORGal. However, first we need to look at how the concepts of autonomy and adaptability relate to the Need.

2.5.3. Autonomy and Adaptability

Being alive is a 24/7 business, and no minute is wasted. Yet we are used to designing equipment with a power-down switch. When you want to use it, you plug it in and you switch it on – when you are done, you switch it off and put it away. Biology does not work that way. Thus, the autonomy paradigm is difficult to incorporate into our traditional design practices. Autonomy requires that a synthetic organism be able to perform more functions over a longer period of time. This effectively increases the uncertainty in the environment, which requires a greater level of adaptability. The more autonomous the synthetic organism needs to be, the greater the level of adaptability required to maintain useful operability. Looking at nature, this comes as no surprise. The architectural 'design' of biological organisms change over a long period of time, perhaps many millennia. During this time it is mostly the same 'design' that operates in almost perpetual manner, generation after generation, with each generation producing the next generation from themselves. This is an extreme level of autonomy, that is, independence from an external force that can help or correct the design. No wonder that biological organisms have a foundation of adaptability.

One can argue that it is not necessary for a synthetic organism to have a considerably long lifetime. However, the IMORGal theory says that in order to keep with the reference design, the best approach is to focus on an individual synthetic organism as having a single lifetime and determine what behaviours it should exhibit during that lifetime, including the time required for adaptation. Switching a synthetic organism on and off (power recycling) is seen as the birth of a new individual. Additionally one could consider power recycling as a form of sleep (requiring the brain state to be saved first in a nonvolatile format) but note that while biological organisms are sleeping, there are a lot of activity going on inside the brain. This activity has its purpose and thus must not be overlooked.

Autonomy can also be called independence, that is, independence from humans. This means that the best situation is to give a synthetic organism a life of its own, that is, a purpose in life which it pursues 24/7 (regardless of the duration of its lifetime) which would place the synthetic organism in the same situation as the biological organism of being always busy with something specific. This does not mean the synthetic organism runs around the whole time but that it is always active, even if that means not consciously performing any action (like sleeping). One could argue the case of service robots where the robot needs only be designed to be a servant to the human and therefore basically either does nothing (awaiting instructions) or carries out an instruction from a human. This sounds sensible at first but let us look at the case of human servants in the age of slavery. The slave was seen and treated as an object or, one can say, as a machine. However, if you set the slave free, it does not just stand there awaiting human orders until its power runs out. It continues to actively fulfil its purpose which is to be human. As said before, it is this purpose that makes the synthetic organism care about something.

The concept of independence does not necessarily mean that each synthetic organism must be able, for example, to recharge its own battery. Depending on the purpose of the synthetic organism, it could be more practical to have a human perform the recharging. At least, while it has power, the synthetic organism will attempt to fulfil its purpose to the best of its ability.

At the end of the day, we are going to have to stop treating these synthetic organisms as machines or equipment but treat them in the same way we treat biological organisms.

As the level of autonomy increases, the synthetic organism needs to be equipped with a larger set of behaviours. I will use the term ethology to refer to this set of behaviours. The Need theory plays the fundamental role in designing the ethology of a synthetic organism.

Adaptability is not an element but a process that acts upon an existing element. Thus, we have our ethology but we need adaptability to optimise the ethology for the current environmental factors being experienced and be able to deal with future changes in the environment. The ethology itself is as though designed for a perfect environment – adaptability is what ports ethology to reality.

As mentioned before, given a set of design parameters whose values cannot be determined a priori, adaptability is simply the ability that is added to firstly determine a value during operation and secondly maintain this value updated in accordance with the current state of the environment. By default, if a change occurs in the environment, the always-active mechanism of adaptability will immediately compensate for it to the best of its abilities. The key element on which adaptability relies is adequate feedback via the sensors. The greater the level of adaptation required, the greater the complexity and intensity of feedback, until consciousness is reached. This feedback and the role its plays in adaptation, given that most adaptation probably requires an immediate response, could possibly be the reason why free-moving biological organisms have a centralised nervous system. Ideally one would like the technologies that apply to the implementation of autonomy and adaptability to be equally efficient.

This concludes the description of the IMORGal theory. The next section continues to describe how we can approach the designing and physical realisation of a synthetic organism.

3. IMORGAL SYNTHESIS

3.1 Design Methodology

The element of synthesis being focused on at the moment is that of realising the synthetic brain. This assumes we already have adequate and known hardware in terms of sensors and actuators, and the desired physical device onto which we would like to realise the synthetic brain. For this situation, synthesis consists of two phases:

- Designing an abstract model of the synthetic brain
- Physical realisation of the abstract design

Note that the abstract model does indeed refer to specific sensors and actuators. What must be excluded from the abstract model is the hardware device or technology in which specifically the synthetic brain will be realised.

In terms of establishing the IMORGal foundation, the idea is to start with the simplest mobile synthetic organism and do a complete design. That is, given the complexity of the sensors and actuators, attempt to reach the maximum level of complexity in the synthetic brain. This will involve extracting all information possible from the sensors and developing the maximum number of behaviours, which will in turn indicate the level of autonomy. What one could expect is that for even a relatively simple set of sensors and actuators, the synthetic brain's design will in comparison be significantly more complex. An effective physical realisation will be crucial.

After such a complete IMORGal has been realised, the complexity of the sensors or actuators will be increased. The design process for a complete IMORGal will be repeated and attempted at first with the building blocks of the previous complete IMORGal. Should these not be adequate, the building blocks can be augmented or revised, or new building blocks could be added. However, the more one can keep the number of different building blocks to a minimum, the greater the eventual extent to which one can apply the same set of building blocks for designing both simple and complex IMORGals.

3.2 Abstract Design

3.2.1. External Environment and Perception

There is more than one way to start an IMORGal design. In the end, all the components (morphology, environment) must be in balance. One approach is to first define the external environment, that is, the environment external to the IMORGal as opposed to its internal environment. The idea is that the design of the IMORGal is done with the external environment as objective. This is because the complexity of the external environment should dictate the complexity of the IMORGal's synthetic brain. In this case, complexity can be seen as the amount of information that can be extracted from the environment by the synthetic brain, which is limited by the complexity of the sensors. The information extraction process is basically the perception of the IMORGal. For an efficient design, it should be avoided that the complexity of the synthetic brain surpasses that required by the environment. This requires additionally that careful attention be paid to the choice of sensors: the chosen sensor should always be the one with the lowest level of complexity that is still able to allow extraction of the desired information from the environment. If it is desired to detect the proximity of objects and an infrared distance is good enough, then this sensor should be chosen above, for example, a CCD camera.

This concept of least complexity forces one to “look through the eyes” of the IMORGal, to “see” the world through its sensors.

Below I have used a different approach since I have already decided to use Roamer, described previously as the one-sensor, one-actuator IMORGal. In this case, the sensor and actuator complexity is fixed and therefore the design is about determining which environment I want Roamer to operate in. Thus, given the complexity of the sensors, actuators and desired environment, I can design with the required complexity for the synthetic brain.

Thus, since Roamer is intended to move around a space and its sensor is one of detecting distances, let us say that the external environment is an enclosed space with a more or less square shape. The surface is flat or has only a slight inclination; the important part is that there are no discontinuities in the surface, for instance, ledges, stairs, holes, and so forth. Throughout this space there are several objects located, some small and others large, some round and others of different shapes like squares and triangles. In terms of manoeuvrability, some areas are wide and open such that it is foreseeable that moving through these areas will be easier compared to other areas where space is restricted. There might even exist a sort of 'door' (a narrow opening) that gives access to a different area of the space.

A further aspect is that all objects are static, that is, they do not change in geometry or location with time. This is opposed to dynamic objects, for instance, a person walking inside the space.

3.2.2. Roamer's Behaviour

How will Roamer behave inside this space? For this example design I have decided on the following: Roamer should explore (roam) the full space available but without making physical contact with any object inside the space or with the boundary of the space itself.

This sounds simple enough but note the following:

- On the one hand, I want Roamer to explore every possible area, that is, every area that allows it to physically pass through. On the other hand, I don't want Roamer to make contact with any objects, which means that I want it to stay as far away from objects as possible. However, exploring the full space would mean having to pass close by certain objects. I don't want Roamer to just stay in one spot and be safe the whole time; such behaviour would not be very useful.
- It makes sense to define a 'safe zone' around Roamer (see Figure 5). If the sensor measures an object that is closer than the boundary of the safe zone, Roamer finds itself in a situation called 'too-close', which means that the probability of Roamer touching that object is unacceptably high. Roamer should react immediately in such a way as to move away from the object until the object is outside the safe zone again.
- Roamer only has the single distance sensor located in front of it, measuring a certain area ahead. The sensor's coverage area is narrow (as indicated in the previous figures). This was specifically chosen because we can make the most of a single sensor if its coverage area is narrow. It allows for a more precise detection of where the object is. On the other hand, this makes life more difficult: facing a certain direction, if the beam is narrow we could have a situation where the measuring beam just misses an object, for example, to the left ahead of Roamer. The sensor says there is nothing ahead but after having moved forward a distance, Roamer's left side would make contact with the object. This is obvious because Roamer has moved in a direction in which it has not seen the full path ahead.

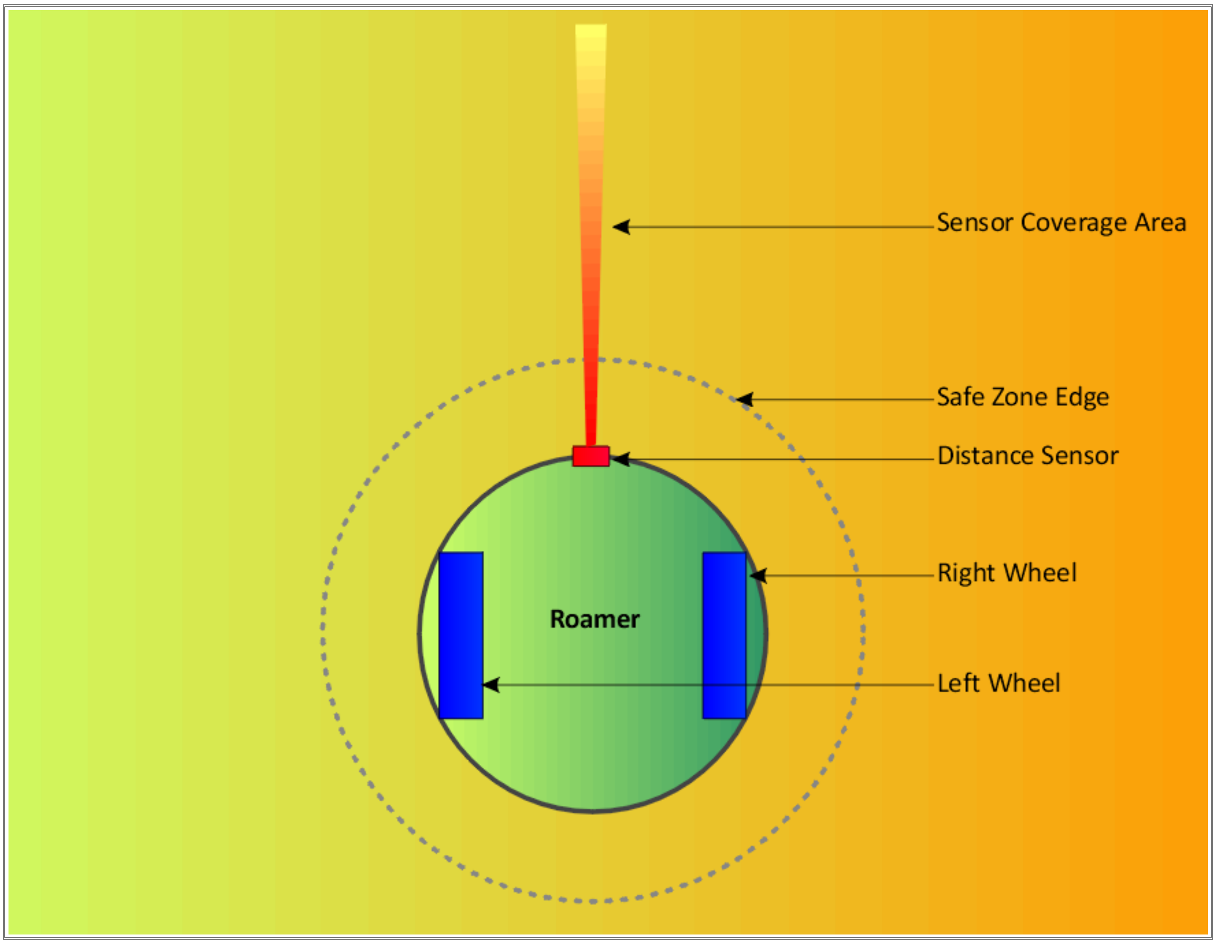


Figure 5: Roamer's safe zone

It needs to move the measuring beam left and right so that the sensor can effectively scan a wider area in front of it. This is indicated in Figure 6.

- Roamer must only move forward and not backward since it has no sensor at the back – it cannot see behind it. Neither must it turn while moving forward – it cannot see to the side. Its only option is to stand on one spot, 'look' to the right and then 'look' to the left – if no too-close object has been detected, the path forward is safe and thus it can move into that direction for a short distance.
- Once Roamer has detected that the area in front of it is safe, it should not be allowed to move forward for too long since it might bump into objects that it could not perceive previously. Compare for example when Roamer moves towards a wall but at a very sharp angle, such that the sensor is unable to detect the object.

Having stated the issues of concern above, let us look at a possible solution for finding the safe zone.

This solution depends on the specifics of Roamer's dimensions and the measuring beam. Firstly, if no too-close object is detected, Roamer moves forward. After a while, Roamer stops, turns to the right on the spot for 45°, turns back left for 90° and then finally turns back again 45° to the right to thus end facing the same direction than its previous forward movement (see Figure 6). Roamer has effectively scanned a 90° sector in front of it. This basic three-phase scanning action is called the safe zone check. Thereafter Roamer moves forward again.

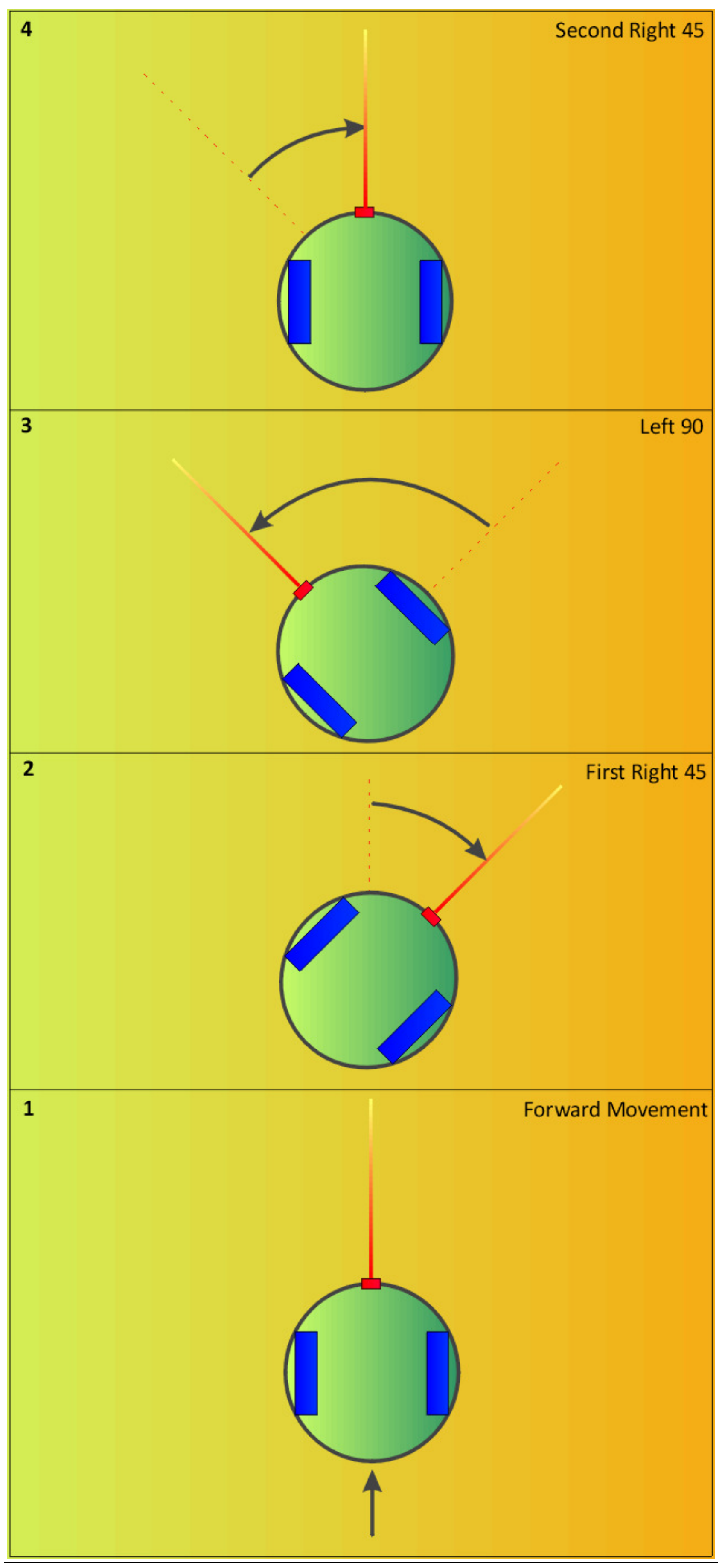


Figure 6: Roamer's forward movement followed by the safe zone check.

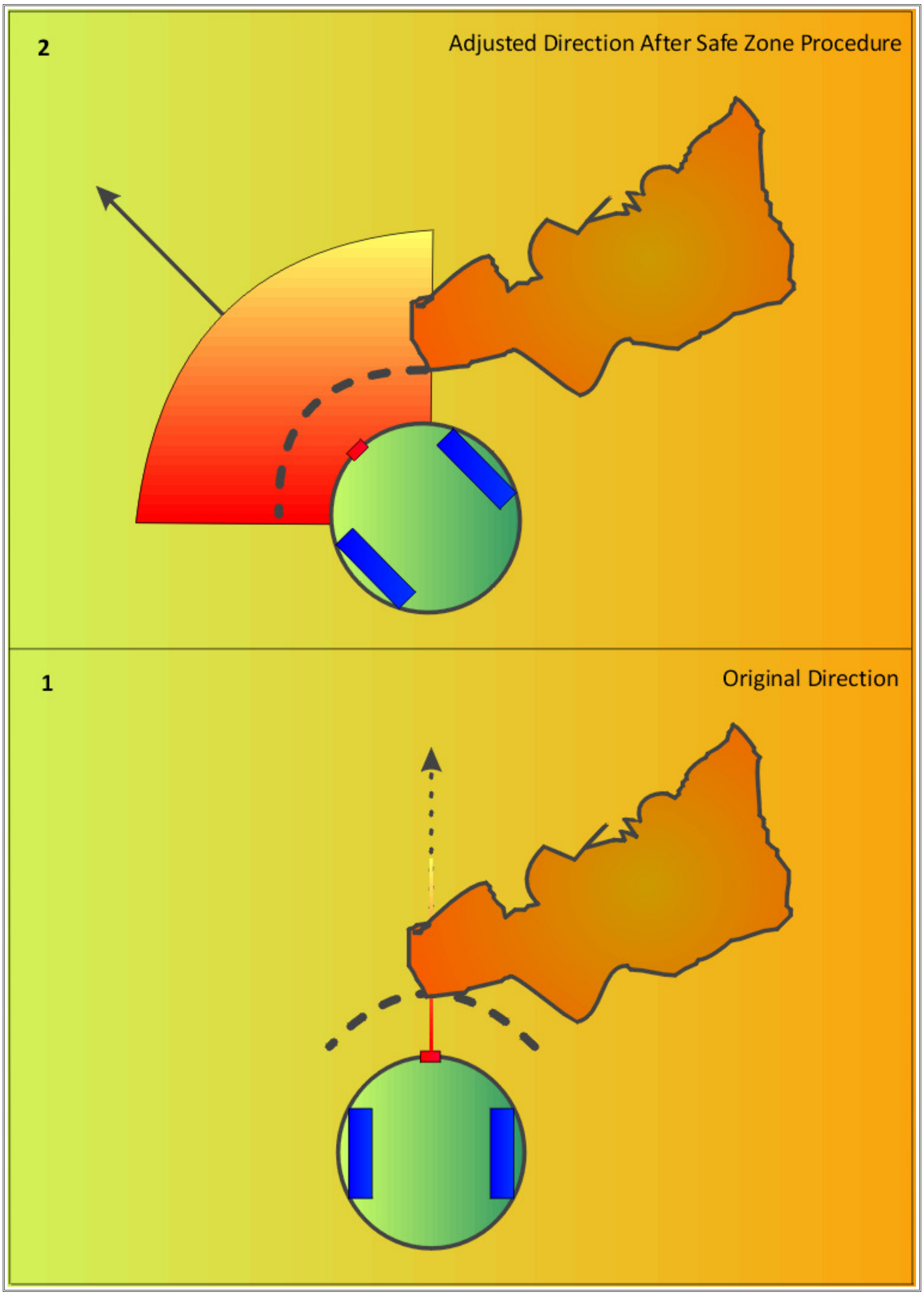


Figure 7: A too-close object detected during forward movement.

Given this pattern of movement, let us look at the possible situations in which Roamer can find itself at the moment of detecting a too-close object. In such situations, the safe zone check becomes a safe zone search – Roamer must now search for a new safe zone. This will require changing the forward movement direction.

- While moving forward. Upon this event Roamer must immediately stop moving forward and start the safe zone search. See Figure 7.
- While turning right 45°. This event starts the safe zone search. See Figure 8.
- While turning left 90°. This event starts the safe zone search. See Figure 8.

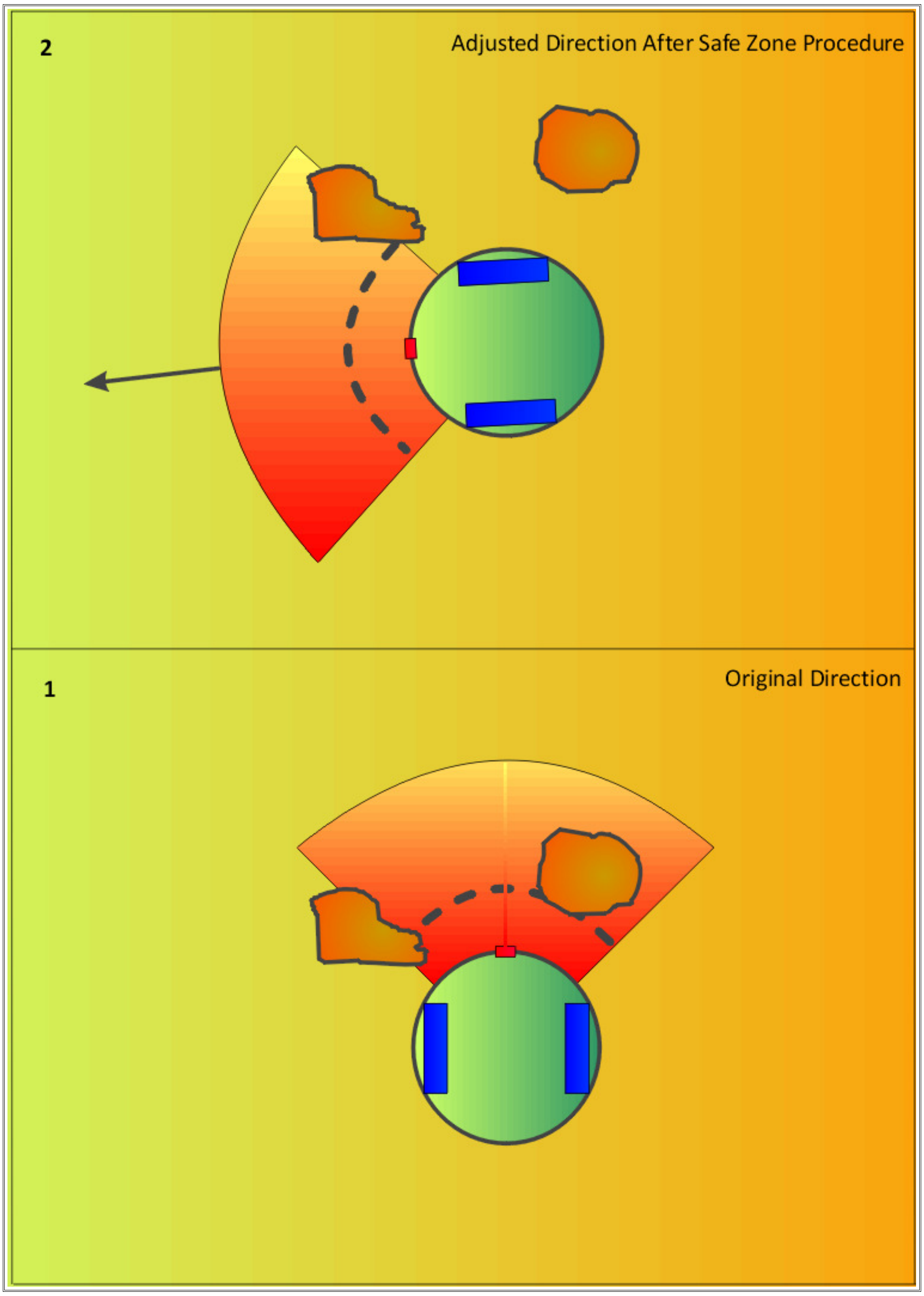


Figure 8: A too-close object detected while turning right and left.

The safe zone search is as follows:

- While measuring a too-close object, immediately start and keep turning left until reaching the point where the too-close object is not too close anymore (thus, the object lies on the safe zone boundary).
- From this point, turn 90° left, followed by 45° right so that Roamer obtains once again an effective 90° sector in which there are no too-close objects.
- If while turning left 90° Roamer measures another too-close object, the safe zone search is restarted.

There are some additional elements I would like to add to Roamer's behaviour. Firstly, I want

Roamer to remember to a certain degree when it encountered the last too-close object and indicate this in its forward movement. Having just performed the safe zone search, the probability is high that it could encounter a too-close object the first time it moves forward again. Thus, there is a larger risk than normal of Roamer making contact with an object during the forward movement before starting the safe zone check. As Roamer goes through consecutive cycles of forward movements with safe zone checks and without any too-close objects detected, the probability of unexpectedly touching an object decreases slightly. Actually, in the situation of lowest risk, Roamer must move its standard distance forward (that is, not influenced by previous too-close objects), with this standard distance signifying a still acceptable distance without the possibility of touching objects because Roamer has moved too far forward before starting the safe zone check. Thus, after having performed the safe zone search, Roamer will be more careful and thus move a shorter distance forward than the standard distance.

The second element is based on the deduction that with Roamer's current safe zone search solution, Roamer has a tendency to always adjust its forward movement direction to the left. This could lead to Roamer not exploring a significant amount of space. Thus, once having adjusted its forward movement direction to the left, Roamer will attempt to turn more than 45° to the right the next time it performs the first step of the safe zone check. This action thus forms a type of direction adjustment towards the right. What one would expect to see is that Roamer, after having past to the left of an object, would change its direction as such that it eventually continues along the direction in which it was moving forward before it encountered the object, albeit with a lateral shift having taken place.

3.2.3. Roamer's Needs

How do we now go from the specification of Roamer's behaviour to designing the Needs for its synthetic brain that would allow Roamer to generate this behaviour?

Here things get complex because at the moment I do not have a simple methodology to suggest for performing such a design. This is due to the newness of the IMORGal technology. However, such a methodology, I believe, will become evident as more and more designs are done. Thus, for now I mention some of the thought processes that I had to follow in order to do the design.

Firstly, the design is strongly graphical. If a software package existed with which to design these brains, it would probably have an appearance similar to LabView's graphical programming interface.

A second remark is that the design is tightly linked to the definition of the required Needs as well as a set of qualitative timing diagrams (similar to Figure 4) that indicate how the inputs, outputs and Value of each Need should change.

Note that this is the abstract design and thus one must be careful not to include details specific to the physical realisation that is to follow. However, the purely abstract design might not contain enough detail, or suggest solutions that are not physically realisable. The idea is to have an accurate enough Need model such that these issues do not arise (or at least not typically) during the physical realisation.

Still, it is acceptable to be forced to revise the abstract design during physical realisation since by itself the abstract design is not physically usable. It is in fact the physical realisation of the design that forms a proof of concept, that confirms the validity and usability of the abstract design.

To start, how do we decide on the type and number of Needs to use? For the moment, I would say to start off by creating a Need for every feature of the design (with 'design' meaning the description of what goes on inside the IMORGal's brain as we did above).

Thereafter, for each Need we firstly need to define conditions that will cause the Need to fire. Remember that by asserting its output the Need is signalling the rest of the brain that it is not currently receiving the expected input to keep it happy (in balance). For this we need to define the inputs to the Need – this is the second part. There are two main types of inputs: those that drive the Need's Value to increase (drive inputs, see Figure 4) and those that affect the rate at which the Value increases or decreases (rate inputs). The drive inputs are necessary for the Value to increase while the rate inputs control the rate at which this increase will occur. The rate decrease inputs come into affect when no drive input is present (as seen in Figure 4). Additionally, the Need can receive an inhibitory input such that a signal applied to this input will decrease or negate the likelihood of the Value ever reaching the condition of being in balance. Furthermore, one can set different increase and decrease rates for the Value, supplied by independent inputs.

A very important aspect is that of scaling. I need to proportionally set the magnitude of the signals applied to inputs of the Needs. This effect is similar to synapses in the brain. The current implementation is to by default have the Need output be the maximum possible size of the Value. At each input of every Need there is a synapse that can scale the input to a smaller value as required by the design. An important limitation that has been imposed by the current Need model is that the synaptic strength can only be set by design and not during operation (referred to here as a nonplastic synapse). Obviously, this infers that no adaptability can be performed that requires a change in the synaptic strength.

By default the Need output will be set to a constant value. However, as we will see below, for some situations the output of the Need is directly the Value of the Need.

To help understand the explanations to follow, have a look first at Figure 9 which shows the completed design.

3.2.3.1. Forward Movement (FM) Need

Let us start with the first Need which I will call the Forward Movement (FM) Need. Its drive input is a signal that says Roamer is currently moving forward. If for some reason Roamer is standing still or doing other movements, there will be no drive input and the Value will decrease until the Need becomes out of balance, upon which the FM Need will assert its output. It keeps the output asserted until the drive input appears again, meaning Roamer is moving forward, and starts increasing the Value. When the Value reaches the condition of being in balance, the Need output is deasserted and the Value starts decreasing again (heading towards imbalance).

3.2.3.2. Too-Close (TC) Need

The Too-Close (TC) Need's purpose is to signal the rest of the brain when Roamer's distance sensor measures a value that indicates Roamer is too close to an object. The TC Need receives as drive input the value currently measured by the sensor. The characteristics of this sensor output value is as described before: it changes according to the measured distance, with larger values indicating objects that are further away. The input is directly taken as the Value of the Need. Thus, for example, as Roamer moves forward towards an object, the measured value (and thus the Value) decreases until reaching the condition of imbalance upon which the Need output is asserted. It is expected that Roamer will respond by performing the actions required to move away from the object. These actions will lead to the Value increasing until reaching the condition of being in balance, upon which the Need output is deasserted.

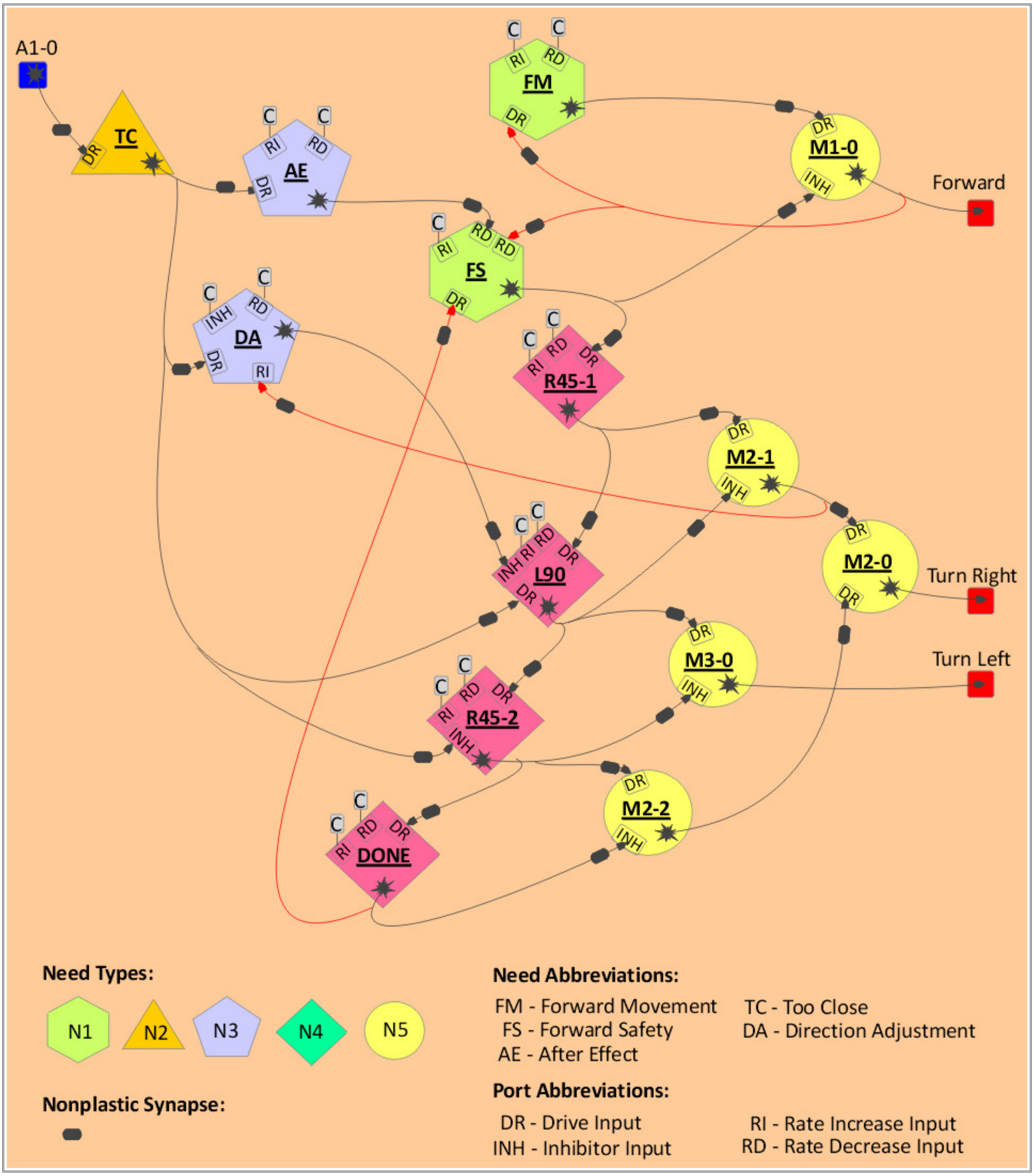


Figure 9: The complete abstract design for Roamer's brain.

3.2.3.3. *After Effect (AE) Need*

The After Effect (AE) Need is the first of two memory-effect Needs such that a certain event will not only have an effect on the Roamer's behaviour at the moment of the event but continue to have an effect for a limited duration after the event has ended. Generally, as more time passes, the 'after-effect' of the event reduces until it has no more effect on the behaviour. The AE Need has to do with prolonging the effect of the occurrence of a too-close situation. The TC Need output itself lasts only while it measures a too-close object. However, the AE Need will have an affect on several forward movements by Roamer after the TC Need stopped firing. As described above, the forward movement effect is that Roamer will move forward for a shorter distance after having just encountered a too-close object, and gradually start moving forward for longer distances between safe zone checks. Additionally I create the AE Need as such that it responds as fast as possible to the TC Need output, meaning, the AE Need fires as soon as possible after the TC Need fires. This will allow me to use only the AE Need output instead of both the AE and TC Need outputs for Needs that require both these inputs. The Forward Safety Need (discussed below) is such as example.

Note that the output of the AE Need is not a constant value but the Value itself. The effect caused by the AE Need must be that of gradual decline.

3.2.3.4. Forward Safety (FS) Need

The purpose of the Forward Safety (FS) Need is to regulate the amount of forward movement and initiate the safe zone check. In essence, Roamer's motion can be divided into two phases: the first is the forward movement which is driven by the FM Need and which should only happen if the FS Need is in balance, thereby indicating it is safe to do so. The second phase is while the FS Need is out of balance (forward movement is unsafe), during which the safe zone check will be performed.

Thus, as part of its regulatory function, the FS Need will inhibit the FM Need output for the duration of the unsafe (second) phase. Additionally, during this phase the FS Need will initiate and drive the safe zone check.

For how long must the FS Need drive the safe zone check? To solve this I create a DONE signal that, when asserted, indicates the end of the safe zone check. Thus, the DONE signal connects to a drive input of the FS Need, driving the FS Need into balance which causes the FS Need output to be deasserted and pass from the unsafe to safe state, thereby allowing the FM Need to move Roamer forward again. The deassertion of the FS Need output causes a chain reaction throughout the safe zone check section, deasserting all Need involved and ending with the DONE signal being deasserted last.

Thus, the drive input to the FS Need is deasserted, causing the FS Value to start decreasing. The decrease rate will determine the period allowed for forward movement until the FS Need becomes unbalanced again and the safe zone check is repeated. The larger the rate decrease input, the faster the FS Need becomes unbalanced. If a too-close object is indicated by the TC Need while Roamer is moving forward, Roamer must stop immediately, which requires the FS Need to fire immediately, which in turn requires the FS Need rate decrease input to be set as large as possible. This value will be supplied by the AE Need output which, as described above, will fire when the TC Need fires. At that instance the AE Need output will be at maximum and remain as such until the TC Need is deasserted. Now the AE Need output starts to decrease: while still large, the FS Need decrease rate will still be fast, causing very short forward movements between safe zone checks. As the AE Need output approaches zero, the forward movement distance returns to its maximum (default) value.

However, once the AE Need output reaches zero the FS Value will stop decreasing and simply remain constant. To solve this, the FM Need output is used. However, not the output directly from the FM Need but the signal after the inhibitory effect from the FS Need has been taken into account (see the M1-0 Need in the figure). This signal connects to the FS Need as an additional rate decrease input. Note that this signal is also zero at times, implying no decrease in the FS Value. However, this is not a problem because during this time Roamer is in any case not moving forward – it is the role of the FS Need to regulate the forward movement and all forward movement will ensure a nonzero decrease rate.

3.2.3.5. First Right45 Need

In the complete safe zone check there are two occasions when Roamer turns right for 45°: once at the beginning of the check and once at the end of the check. Remember that the basic sequence is Right45-Left90-Right45. Here I discuss the first 45° right turn – the second follows the discussion of the Left90 Need.

The first Right45 (R45-1) Need is directly driven by the FS Need output and marks the start of the safe zone check. Let us define that the output of the R45-1 Need drives the wheels such that Roamer turns right by 45°. The principle difference of this Need to those mentioned before is that the R45-1 Need functions in reverse, that is, its conditions of being in and out of balance are reversed. This Need is in balance when no input is received and thus does not assert its output regardless of the amount of time that has passed. As soon as the Need receives an input signal, its Value starts to increase until reaching a state of imbalance, upon which it asserts its output. This output will be maintained asserted until the input signal is deasserted at which point the Value will start decreasing.

Note that it is the R45-1 Need output that drives the right turning action: its drive input is the FS Need output, which means that when the FS Need fires there is a certain delay until the R45-1 Need fires since the R45-1 Value must first increase and reach imbalance. In terms of Roamer's behaviour, this delay is the delay between Roamer stopping to move forward and Roamer starting to turn right. Since Roamer should move forward slowly (meaning, minimal momentum), this delay can be set as small as possible.

In addition to driving the right turning action, the R45-1 Need output is also connected as a drive input to the Left90 Need.

3.2.3.6. Left90 Need

Thus, the Left90 (L90) Need receives as drive input the output from the R45-1 Need. The output of the L90 Need once again will cause the left turning movement of Roamer. However, since the R45-1 output is causing Roamer to turn in the opposite direction, the L90 output must additionally inhibit that right-turning movement so that the left-turning movement will occur unhindered.

The important aspect here regarding the turning Needs is that of time. While the R45-1 Need is firing but not the L90 Need, Roamer is turning right. Given the angular velocity applied to the wheels and defining the increase rate of the L90 Value, we are determining for how long the right-turning will happen. As soon as the L90 Need fires, the right-turning stops and the left-turning starts. By configuring all the relevant parameters one can set the eventual amount of degrees that Roamer will turn right. The same also applies to the L90 and second Right45 Needs. Along the chain of turning movement Needs, as soon as a post-Need fires, the pre-Need's turning movement ceases.

For the L90 Need we must additionally take into account the Too-Close (TC) Need output. That is, while the R45-1 Need is firing (Roamer is turning right), at the moment the TC Need fires (meaning Roamer is detecting a too-close object) Roamer must immediately stop turning right and start turning left until reaching the edge of the safe zone (the TC Need stops firing) – from that point Roamer must turn 90° left without encountering any too-close objects. To do this we set the TC Need output as a large drive input such that when the TC Need fires, the L90 Need fires immediately thereafter which causes Roamer to stop turning right and start turning left. Note that if Roamer detected the too-close object while moving forward, the same mechanism will cause no turning to the right since the L90 Need should fire before the R45-1 Need. The rest of the turning behaviour is determined by the second Right45 Need. Before discussing the Left90 Need, let us look at the second memory-effect Need (the first being the AE Need) which has an additional influence on the performance of the L90 Need.

3.2.3.7. Direction Adjustment (DA) Need

The Direction Adjustment (DA) Need is similar to the AE Need. Here the prolonging effect is used to give Roamer a correction-to-the-right feature, that is, after having had a too-close event, during the first right-turning movement (R45-1), Roamer will turn more than the usual 45°. This is because Roamer's too-close event handling mechanism causes Roamer to always pass the object to the left – that is why we increase the amount of right turn performed by the R45-1 Need.

The AE Need requires the too-close effect to last multiple safe zone checks whereas the DA Need's effect should only have an affect the next time the R45-1 Need is turning Roamer. This method of direction adjustment obviously ignores elements such as the amount of left-adjustment caused by the too-close object with respect to the direction in which Roamer was moving forward just before detecting the object.

The too-close effect for direction adjustment is activated when a too-close object is detected while the R45-1 Need turns Roamer to the right. This effect is basically the gradually-declining output of the DA Need (as with the AE Need output) and connects to the inhibit input of the L90 Need. This inhibition has the following effect: the larger the output from the DA Need, the stronger the inhibition on the L90 Value, the longer it will take for the L90 Need to reach imbalance, and thus the longer the period that Roamer turns right due to the R45-1 Need. The inputs to the DA Need effectively form a logical AND function: if only the TC Need fires or the R45-1 Need turns Roamer, the DA Need should not have an effect on the turning behaviour. However, when both those conditions are true, the DA Value increases. The increase rate is not designed specifically and is simply set as fast as possible. Thus, the amount of direction adjustment will be more or less constant and, although not precise, should give Roamer a correcting-to-the-right inclination after passing to the left of an object.

Needless to say, there are many different ways to consider and implement different forms of direction adjustment. More complex Need designs can generate more complex and specific direction adjustment.

3.2.3.8. Second Right45 Need

As with the two previous turning Needs, the main purpose of the second Right45 (R45-2) Need is to inhibit Roamer from turning any further left, and to start turning right again. As before, the amount of degrees that Roamer will turn left is determined by the time it takes the R45-2 Value to reach the condition of imbalance, that is, the period between the L90 Need firing and the R45-2 Need firing. By setting the R45-2 Need parameters appropriately we can ensure that Roamer turns more or less 90°. However, the R45-2 Need must take into account the state of the Too-Close (TC) Need, that is, while the TC Need fires, the R45-2 Value must be set to and held at zero. When the TC Need stops firing, Roamer is on the safe zone edge and from this point Roamer must turn 90° left which is when the R45-2 Value must start increasing. Thus, the TC Need output is connected to the inhibit input of the R45-2 Need.

There lacks a final post-Need to regulate the amount of degrees that the R45-2 Need turns right. This need is called the DONE Need.

3.2.3.9. DONE Need

The output of the R45-2 Need inhibits the any additional left turning and drives Roamer to turn right. It further drives the DONE Value increase which is configured such that Roamer will have turned 45° right when the DONE Need fires. The DONE Need output has two purposes: firstly, the usual of inhibiting the right-turning action caused by R45-2, and secondly to serve as the DONE signal that indicates the end of the safe zone check to the Forward Safety (FS) Need. The next movement thereafter should be Roamer moving forward.

3.2.4. Discussion

Notice that in Figure 9 there are 5 different types of Needs. The Needs of a particular type are located next to one another in the figure. This format encourages the formation of different regions in the artificial brain with each region containing a similar configuration of the basic element which is the Need. This idea comes from the presence of different regions in the biological brain: each region consists of neurons in principle but the neurons from one region are slightly different from neurons in other regions.

As well, notice the connections in Figure 9 with the colour red. These connections indicate feedback paths and are highlighted because in conventional digital design incorrect feedback can have a drastic effect on the design's functioning. However, it is well known that feedback connections are common in the human brain and even outnumber feedforward connections in areas such as the visual processing regions of the brain. Thus, the idea is not to discourage the use of feedback in IMORGal designs but to be aware of the problems that they cause in conventional designs. An additional point to consider is that whether these feedbacks should be internal or external to the synthetic brain.

More aspects regarding the current abstract design methodology are discussed in the final section of this document.

3.3 Physical Realisation

Let us now take the abstract design and implement it physically. Here I must admit that I always had the FPGA (Field Programmable Gate Array) in mind when I thought about the IMORGal concepts. The FPGA is the only electronic device I considered initially for physical realisation because, since an FPGA by default consists of a large amount of configurable logic, it permits implementing hardware nodes that execute in parallel in much the same way as found in the biological brain. The complete device is basically a layer of programmable generic low-level logic elements in quantities from thousands to hundreds of thousands. Nowadays, most FPGAs have embedded multipliers which in this case were specifically used for Value update calculations. Note that with an FPGA you do not write software but rather describe hardware, which involves configuring the logic elements to form certain logic functions.

The biggest disadvantage is that an FPGA consumes much more power than a typical embedded microcontroller. This has practical implications that must be taken into account when designing the IMORGal's ethology, lifetime and external environment.

It could happen that the signals between the FPGA and the sensor and actuators (I/O signals) are physically incompatible (for example, incompatible voltage levels). This could require the use of a microcontroller to perform the necessary signal conditioning such that the specific physical requirements can be met for each element (FPGA, sensor and actuators). The physical realisation starts by performing FPGA simulations. Due to the hardware nature of the FPGA, these simulations can exactly show what the resulting behaviour of the FPGA-realised synthetic brain will be, given that the sensor input signals can be sufficiently simulated. This is made difficult by the fact that Roamer itself, by driving the motors, affects the value measured by the sensor.

The hardware description is done with the VHDL programming language which is, together with Verilog, the most widely used hardware description languages for an FPGA.

I follow the following approach for the physical realisation of the abstract Need model:

- First I show a generic VHDL model, meaning, this model is written in VHDL but without targeting a specific FPGA device.
- I perform various functional simulations to verify that the physical design matches the abstract design.
- I perform FPGA synthesis using FPGA tools that will implement the VHDL code on a specific FPGA device. Statistics from this process show how much FPGA resources are consumed by the current VHDL code (indirectly speaking, by the synthetic brain).

3.3.1. VHDL Implementation

First it is important to note that the rate at which changes occur inside the synthetic brain is not in the nanosecond range but the millisecond range (as also found in the biological brain), at most hundreds of microseconds. Immediately the main advantage due to this is that there are no timing issues as found in conventional FPGA designs that work normally in the order of 3 to 100 nanoseconds. Therefore I do not mention any timing aspects at all during the rest of the design description since these are practically irrelevant.

The type of digital design applied is that of synchronous design, meaning, all states inside the FPGA is updated at fixed time increments instead of continuously. The complete design is thus based on a single clock signal that is received by all logic elements implemented. At the clock edge, each state changes to its next state according to its inputs at that moment. These changes propagate across the complete FPGA as per the design of how logic elements are interconnected.

For the simulations below a clock rate of 10 kHz is used, a value so low that it is unheard of in conventional FPGA designs. This aspect of a significantly slow clock rate is one of the most important advantages of the IMORGal brain when compared with other synthetic brain realisations. For one, this clock rate implies that different technologies than the high-performance CMOS could be considered. Another is that such a slow clock rate significantly reduces the dynamic power consumption of the device, with power consumption being a serious problem as mentioned before, especially in designs with the larger FPGA devices.

Figure 10 shows the VHDL entity used for the Need model, which includes a separate VHDL entity that forms the Value of the Need. Much of the VHDL model correspond closely to the abstract model and thus the components of the VHDL model will be clear from the start. The main detail is how the Value's rising and falling are implemented. For this, a single-order auto-regressive filter with positive feedback is used, which consists of a single multiplier, adder and accumulator.

The data format is based on an unsigned binary fractional format denoted here as [1].[17], meaning that data in this format has 1 decimal bit and 17 fractional bits. The fractional part also determines the resolution of the data in decimal notation. Since the value is unsigned, the decimal range of this format is from 0 to 2.0 – (2**-17). However, to simplify arithmetical computations in the model, certain signals inside the Need have a decimal range less than [1].[17]. These different formats are indicated by different colours in the figure. Note that all signals have the same resolution of (2**-17).

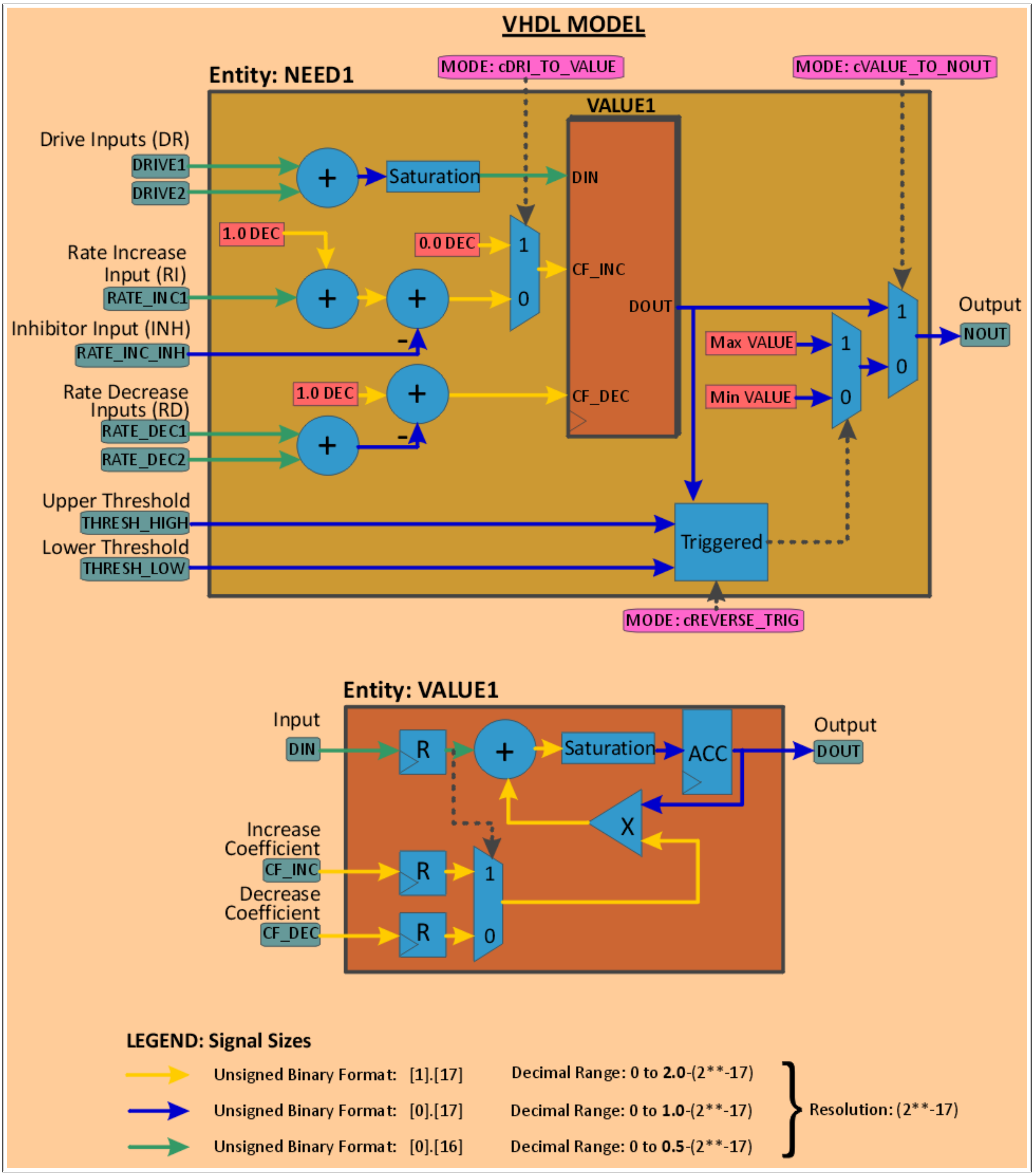


Figure 10: The VHDL description of the Need.

The format of [1].[17] was chosen to correspond with the dedicated 18x18 multipliers found in certain FPGAs. Multipliers of size 9x9 and 36x36 are also normally available in those FPGAs but 18x18 was finally chosen due to being a good trade off between size and range issues. Regarding size, the larger the multiplier, the more FPGA resources are consumed per Need which decreases the maximum number of Needs one can implement per device. Regarding range, it is important to note that the difference between the maximum and minimum time required by the Value to pass between its maximum and minimum values is determined by the data format of the Value. The current implementation allows the Value to have a binary range of 0 to 131071 (that is, 2**17-1).

As indicated in the figure, the Value is updated every clock cycle and thus the maximum time (or range) for the Value to change between its maximum and minimum binary values occurs when the Value increment or decrement is linear, which means the Value is simply a counter. Thus, the maximum time is equal to 131071 clock cycles. Obviously, the larger the clock period, the slower the clock rate, the larger the range but the slower the Need is able to change its Value. Note that this maximum time is for when the threshold levels are set to the extreme values (largest and smallest) of the Value. The minimum time was found to be typically 1 to 2 clock cycles.

The concept of Value range must be understood as follows: given a clock cycle of 100 us (10 kHz), the Value range is around 13,1 seconds. This means that if we have a situation of a drive input being present and the Value being at maximum value, then when the drive input ceases and the decrease coefficient is at its minimum value and the thresholds at the extremes, the Value will decrease and reach zero after around 13,1 seconds at which the Need will change from balanced to unbalanced and thus assert its output. This is the maximum time that can pass between the drive input ceasing and the Need firing. Thus, for the After-Effect Need in our design, the after-effect can only have an influence for a period of around 13,1 seconds.

The designed VHDL entities are made configurable to a certain extend, as indicated by the MODE parameters in the figure, such that there is only one Need entity but which can be configured to implement the different Need types mentioned in the previous section. This aspect allows for a more compact VHDL design. These modes come from the abstract design and are as follows:

- cDRI_TO_Value: This mode sets the Value equal to the value of the first drive input
- cVALUE_TO_NOUT: This mode sets the output of the Need equal to the Value
- cREVERSE_TRIG: This mode allows the reverse functioning of the thresholds, as described above for the R45-1 Need

The configurability of the Need allows for better optimisation during synthesis such that less device resources are used.

A basic aspect of the Need inputs is that a larger value means a greater effect, which provides a more intuitive feeling when designing these inputs. Furthermore, note that either larger drive inputs or larger rate increase inputs will cause the Value to increase faster. A small value as a drive input or rate increase input will not guarantee a slow increase of the Value. Both must be small for that purpose. The inhibitor input is implemented as part of the rate increase input mechanism of the Need. This approach was chosen to correspond with the Value implementation.

As mentioned before, part of the design involves scaling of the Need inputs and thus in the simulations to follow one can see a lot of variation from Need to Need regarding the size of the inputs applied. To facilitate the scaling, the method of binary shifting is used, meaning, that the the input is scaled down by shifting its value to the right by a certain amount of bits before connecting the input to the Need input port. This simplifies the overall design effort but is more restrictive regarding finer adjustment of the inputs. The shift function is implementation in standard logic and thus do no require the use of a multiplier.

In Figure 10 one can additionally see different configuration ports for the Need to allow the Need to operate in different modes.

3.3.2. Functional Simulation

Below are shown the simulation results done for each Need as described in the previous section and which strive to further help explain the current design that is being implemented. All simulations are functional simulations done with ModelSim.

Figure 11 shows the Forward Movement Need. It provides a graph similar to the abstract Need. The thresholds are set by default equal to the Value maximum and minimum values.

Figure 12 shows the Too Close Need. Appropriate stimulation for the sensor output is generated via the VHDL test bench. The thresholds are different from the default values.

Figure 13 shows how the Value of the AE Need is also the output of the Need.

Figure 14 shows a more complex simulation involving the Forward Safety Need. The short pulses on DRIVE2 is the DONE signal that comes from the DONE Need. The first of these pulses causes the FS Need to get back in balance and forward movement (RATE_DEC1) continues for the default time. This is followed by a safe zone check. When Roamer moves forward again, it detects a too-close object (the output of the AE Need at RATE_DEC2) and immediately initiates the safe zone search. For the next two forward movements, the time is shorter than the default due to the effect of the AE Need. Thereafter the forward movement continues again for the default time.

Figure 15 shows how the period for drive-input-nonzero to Need-firing is set very short since this Need is not involved in the timing of the right and left turning movements.

Figure 16 however does show how that period is now around 1 second, and is meant to indicate 45 degrees of turn. During the second safe zone check, DRIVE2 indicates that a too-close object was detected and this causes the Need to fire almost instantaneously. The following safe zone check shows how the period for drive-input-nonzero to Need-firing is slightly more than 1 second – this is the direction adjustment to the right due to the too-close object detected during the previous safe zone check and inputted to the Need via the RATE_INC_INH port that is the output of the DA Need. The final safe zone check shows a normal right turning movement of 45 degrees again.

Figure 17 shows the AND function implemented by the Direction Adjustment Need. That is, only when the Too Close Need fires and Roamer is busy with the first right-45 turning movement does the Value reach its maximum.

Figure 18 shows the second Right-45 Need. The period for drive-input-nonzero to Need-firing is about one second and a half, and is intended to cause left turning by 90 degrees. During the second safe zone check one can see that the Value does not increase while the TC Need output is asserted.

Figure 19 represents the behaviour for all the motor Needs (M1-0, etc). Their behaviour is very simple and one can see how the Need output corresponds closely to the drive input except when the inhibitor input is asserted externally.

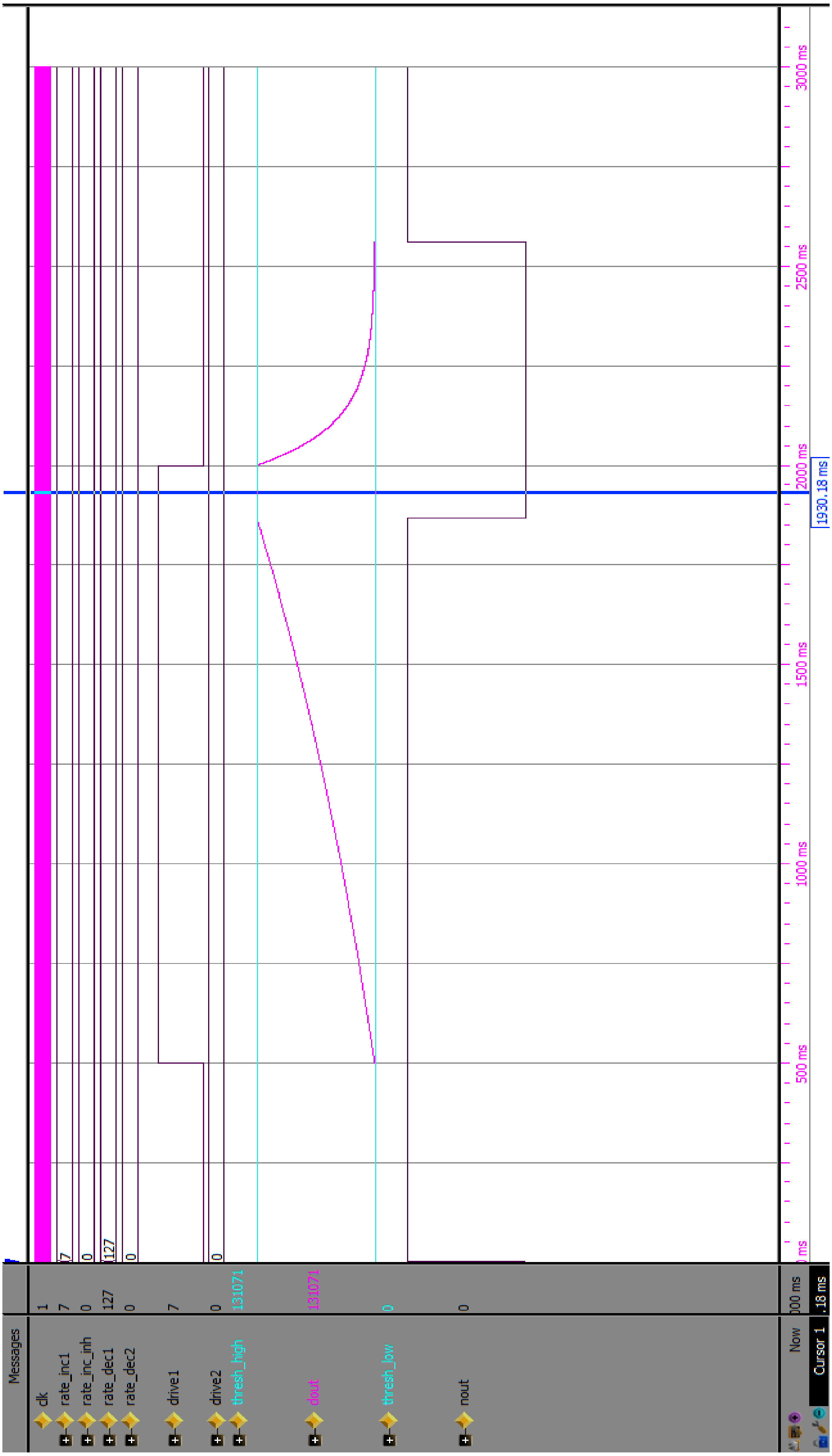


Figure 11: Simulation result for the Forward Movement Need

Figure 12: Simulation result for the Too Close Need

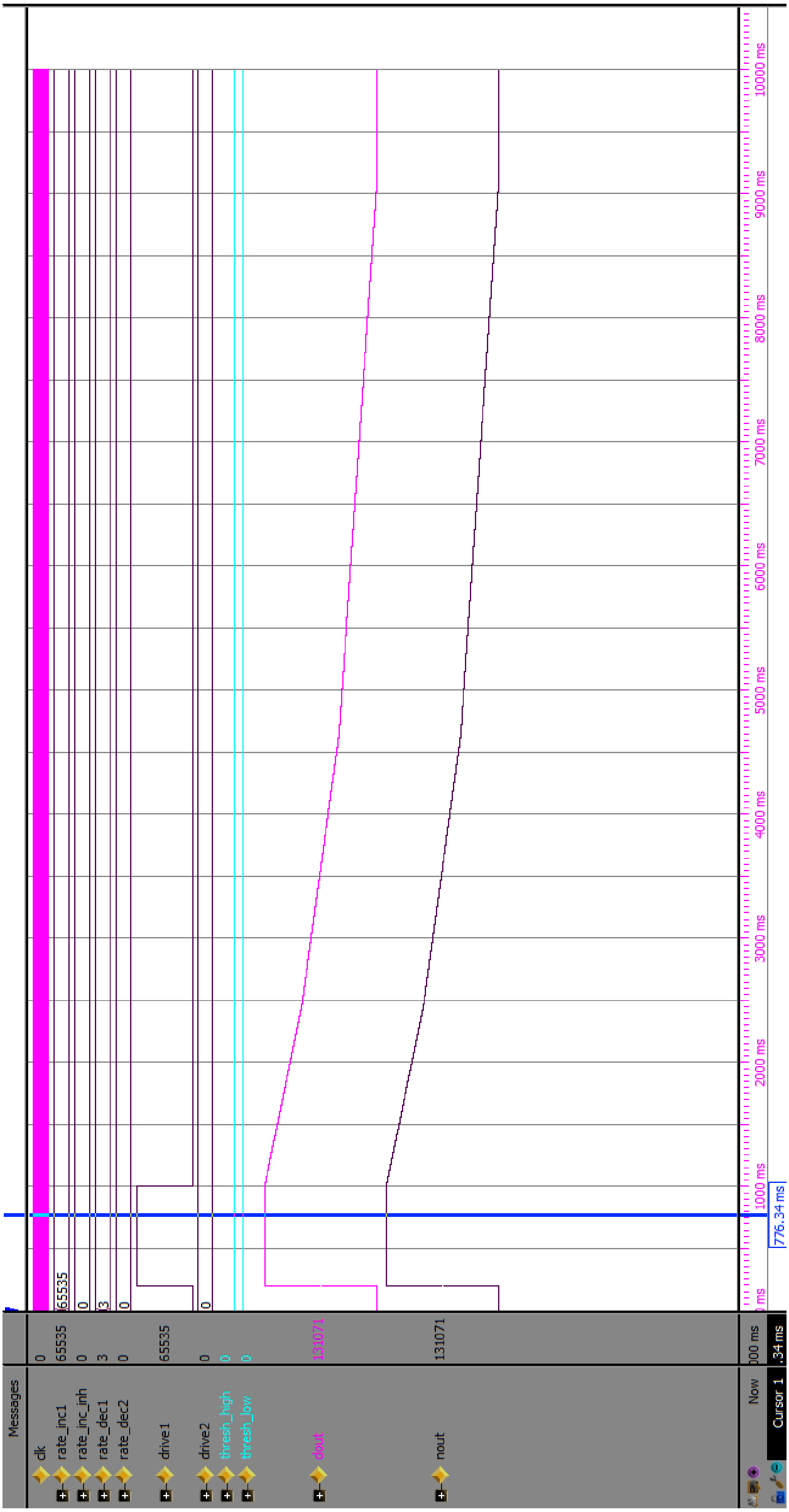


Figure 13: Simulation result for the After Effect Need

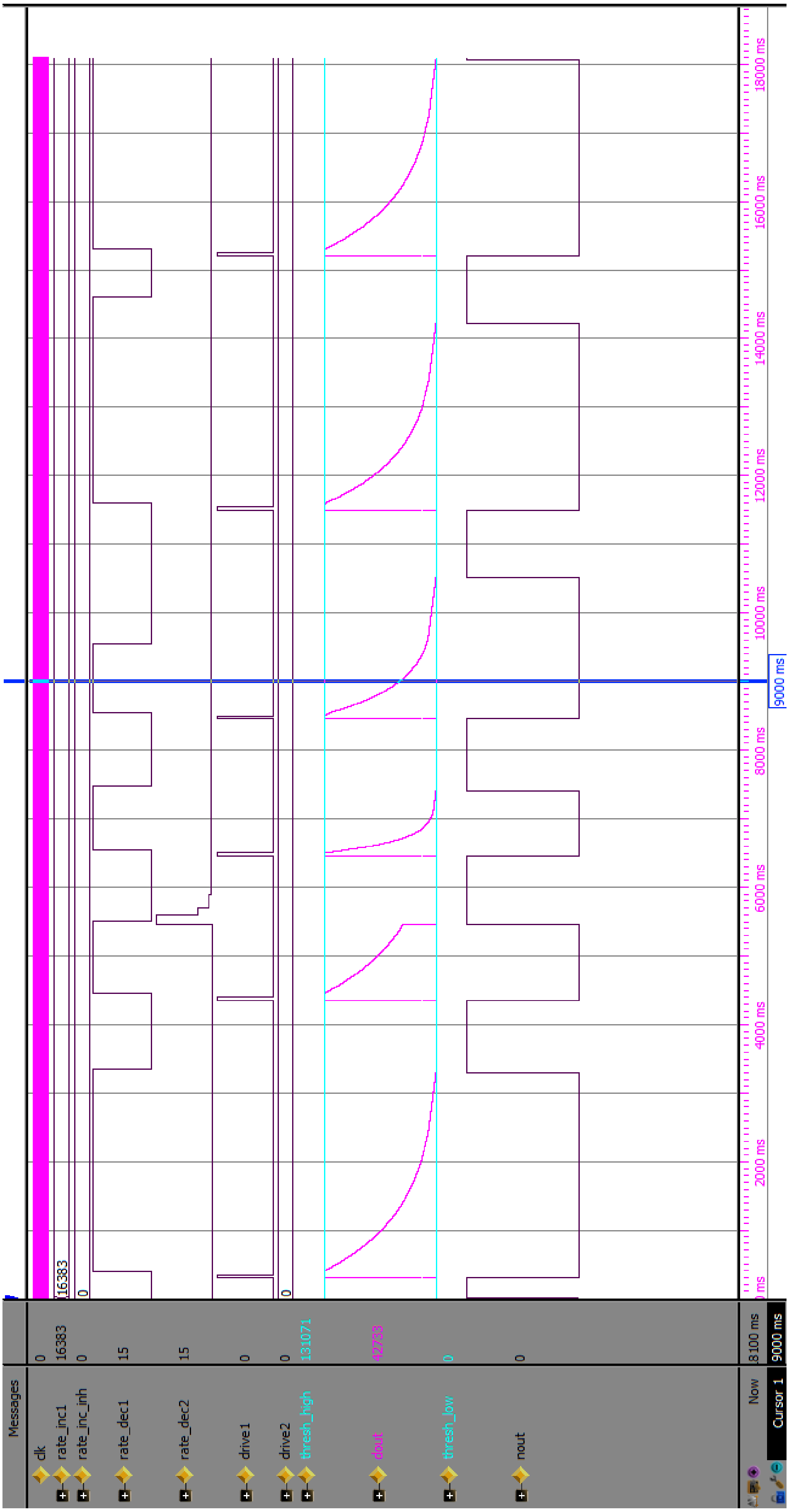


Figure 14: Simulation result for the Forward Safety Need

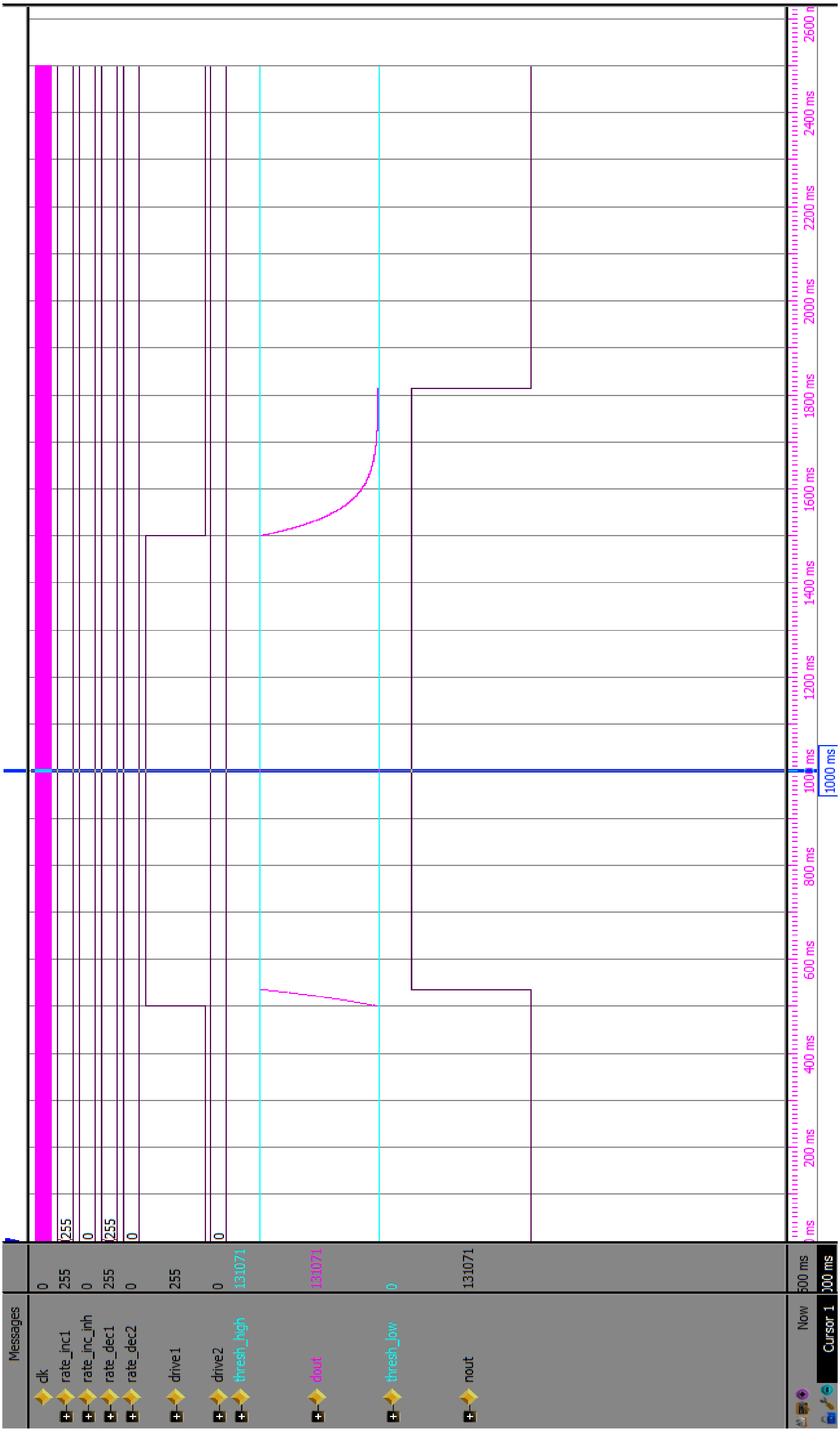


Figure 15: Simulation result for the First Right-45 Need

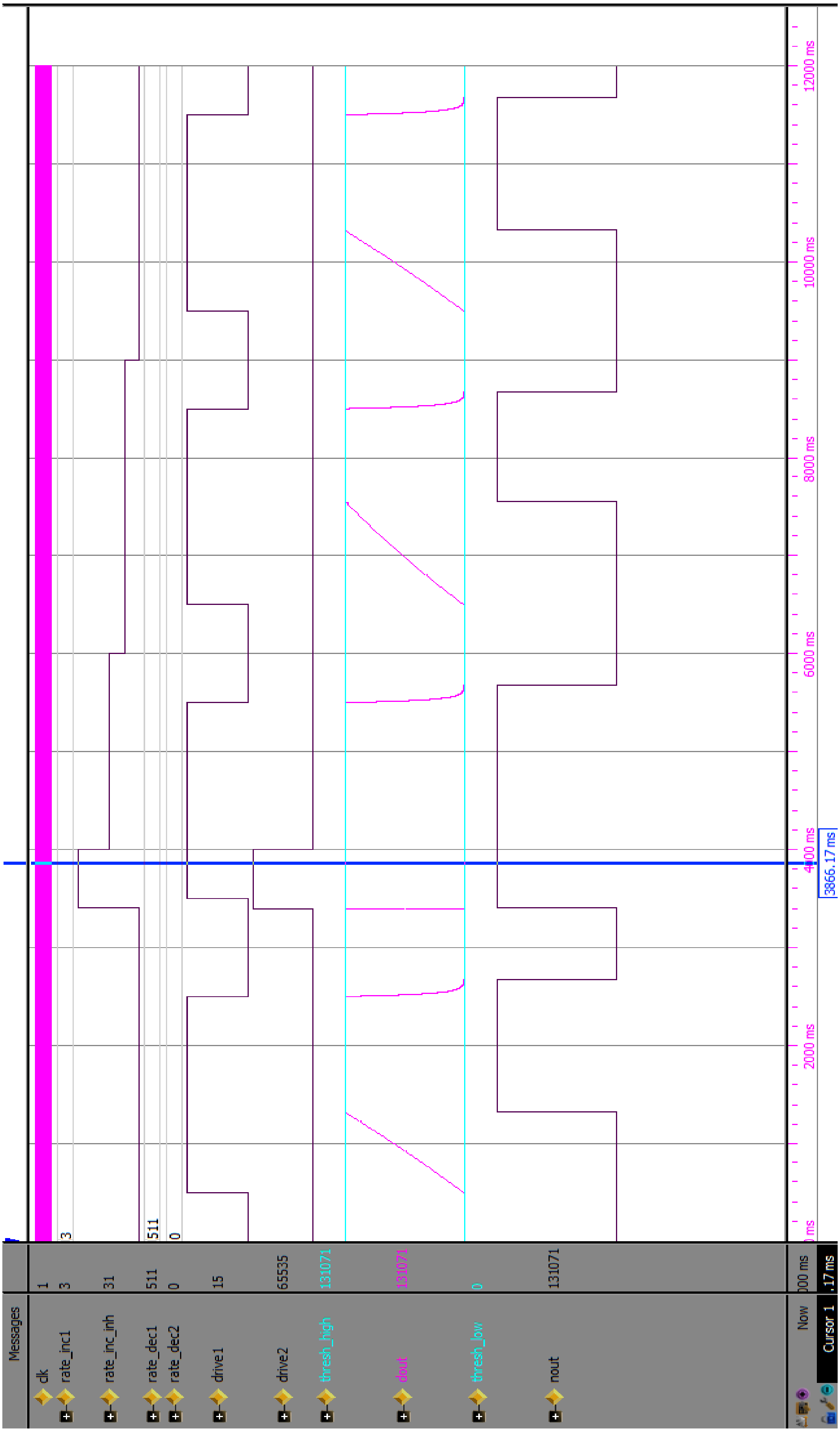


Figure 16: Simulation result for the Left-90 Need

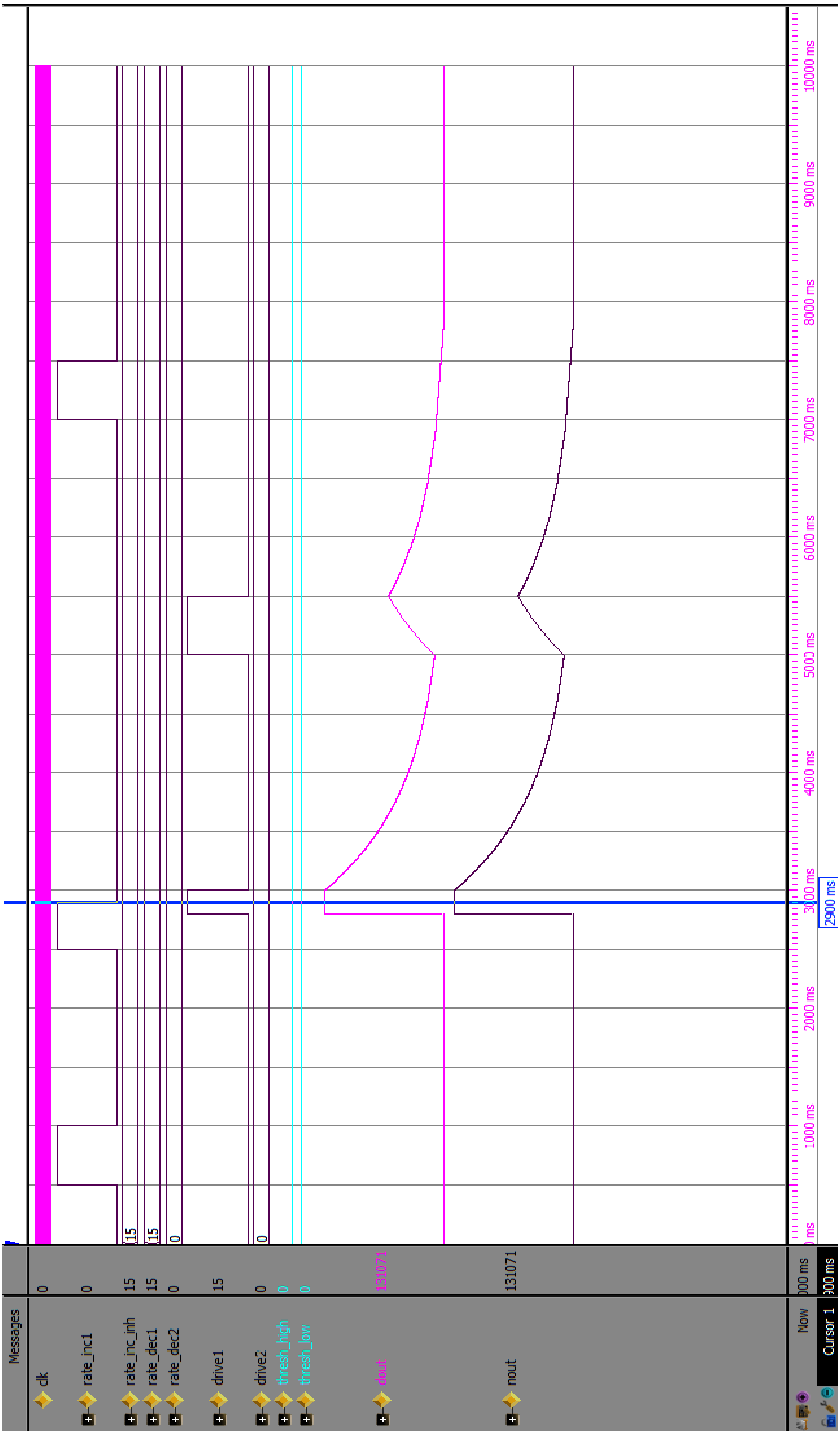


Figure 17: Simulation result for Direction Adjustment Need

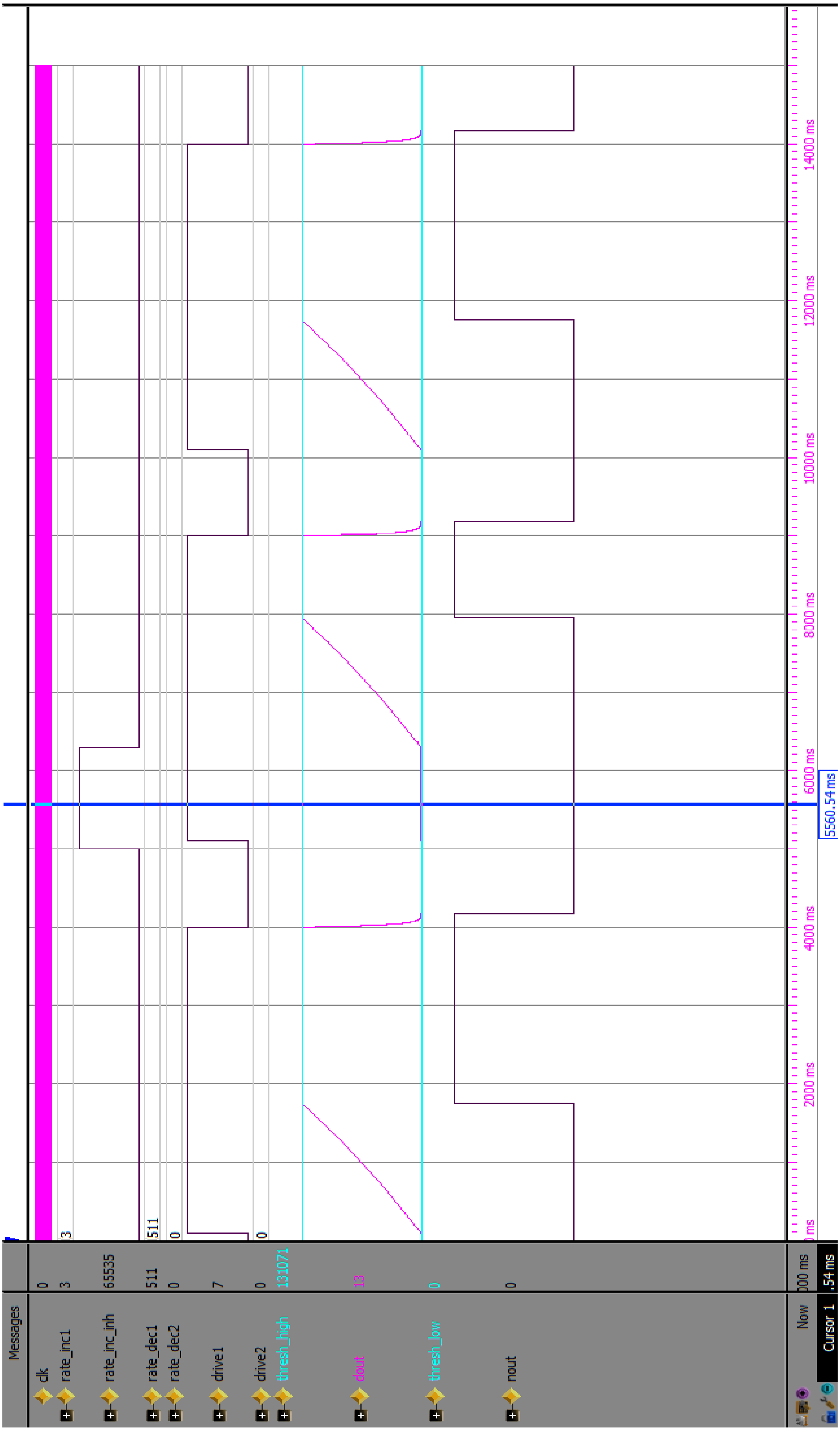


Figure 18: Simulation result for the Second Right-45 Need

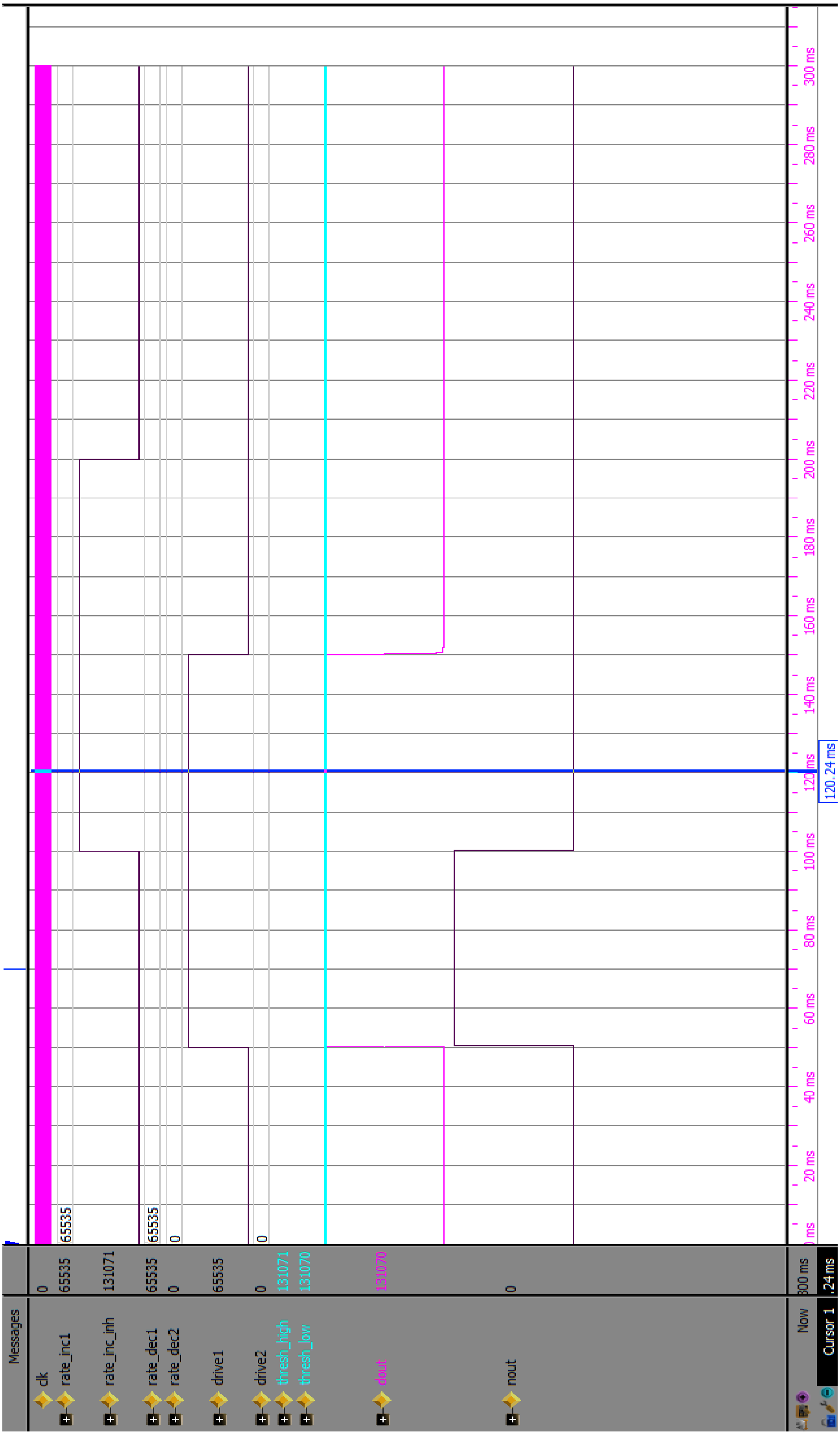


Figure 19: Simulation result for the motor-type Needs

Now that I have done a separate functional simulation for each Need, I create a single main VHDL file at the top of the hierarchy that interconnects all the Needs together. Instead of supplying Need input stimuli for the simulation and observing the Need's Value and output, I now work with the organism interfaces, that is, the distance sensor input and the movement outputs, as indicated in Figure 9 with blue and red squares, respectively. The only value I stimulate during the simulation is that measured by the sensor and supplied to the brain.

The scaling of Need inputs were kept initially the same as for the separate simulations. I am happy to say that simulation of the final design required very few parameter adjustments, including the scaling effort. This shows that the approach of first simulating the Needs separately is valid and especially important to simplify the design verification, that is, that the VHDL design matches closely with the abstract design.

Thus, for the complete brain design in VHDL I show 5 simulations below. For clarity I display only the organism input and output value changes instead of all Need Values and outputs.

In Figure 20, the first simulation shows 30 seconds of brain activity from power-up without encountering any too-close object during the simulation. Notice that since there is no reset signal, Roamer starts acting immediately after power-up according to its design. Thus the initial behaviour is determined by the power-up values of registers inside the FPGA. Quite intuitively, just as a baby comes screaming into the world, the first action by Roamer is to perform a safe zone search by which it immediately halts the first Right45 movement, starts turning left in search of a safe zone, and then completes the search with the second Right45 movement. Now Roamer feels safe for the first time and thus starts moving forward.

Following this we see twice Roamer only moving forward for a small period before performing the safe zone check. This is due to the after-effect of the initial unsafe situation at power-up. Now the after-effect's influence is over and we see two normal situations of Roamer moving forward the maximum distance before performing a safe zone check again.

Figure 21 shows the second simulation which is a repeat of the first simulation except that I cause a too-close object detection during forward movement after about 15 seconds. What we see is that the first Right45 movement is not even performed, followed by the Left90 movement waiting for the end of the too-close situation (the edge of the safe zone), following by the Left90 turning its usual 90 degrees, followed by the second Right45's normal movement. Roamer feels safe again and now moves forward for a small period until repeating the safe zone check. Here we can see the direction adjustment being performed in that the first Right45 movement last longer than usual. As no additional too-close objects are detected, the forward movement length returns to normal.

Figure 22 shows the third simulation. Now the too-close object is detected while performing the first Right45 movement. Turning right is immediately stopped, followed by the usual safe zone search. Once again the direction adjustment is present.

Figure 23 gives the results of the final simulation which shows the situation of too-close object detections during two consecutive first Right45 movements, followed by the direction adjustment.

Figure 24 simulates too-close objects during the first Right45 and Lef90 movements within the same safe zone check.

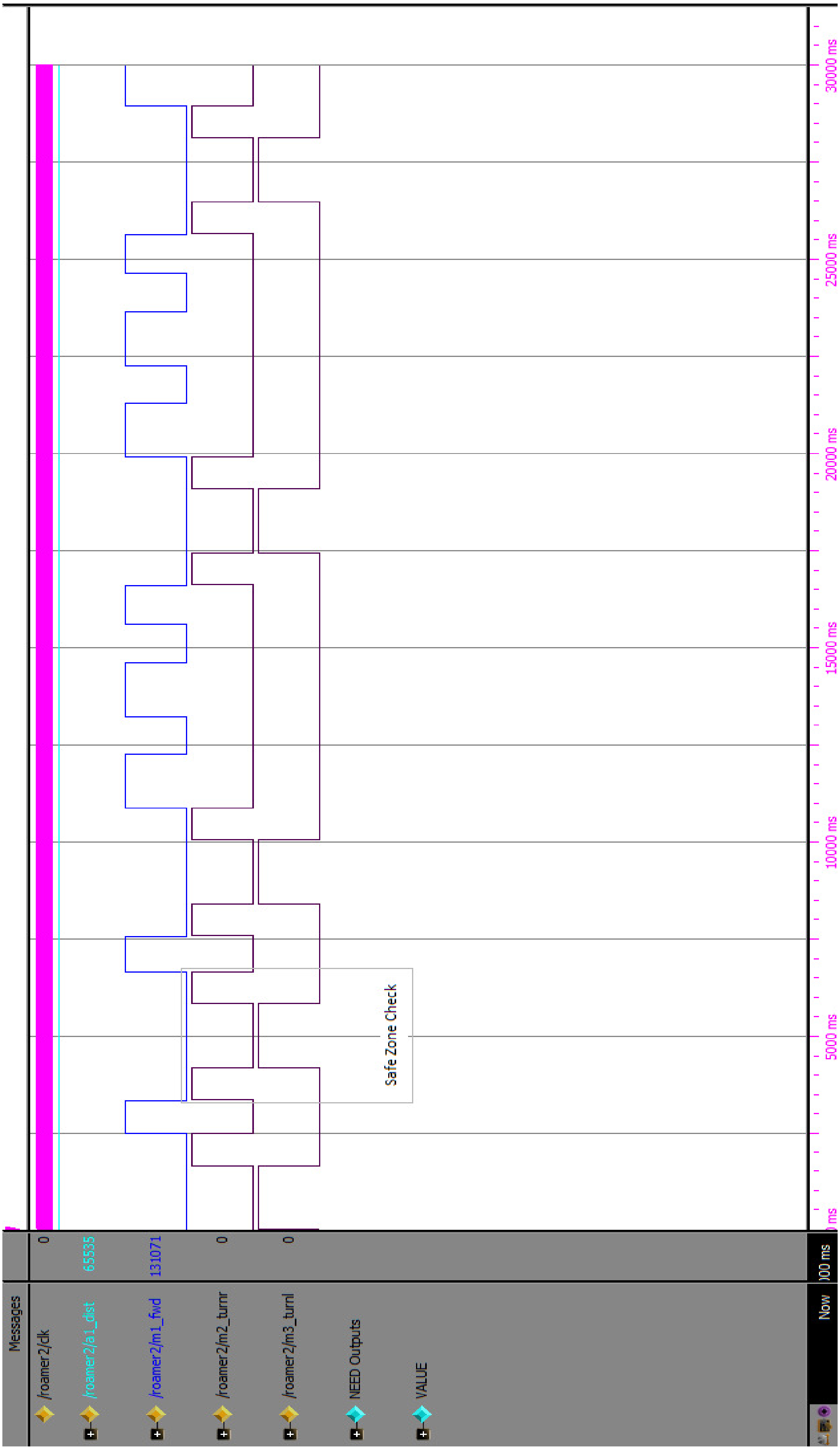
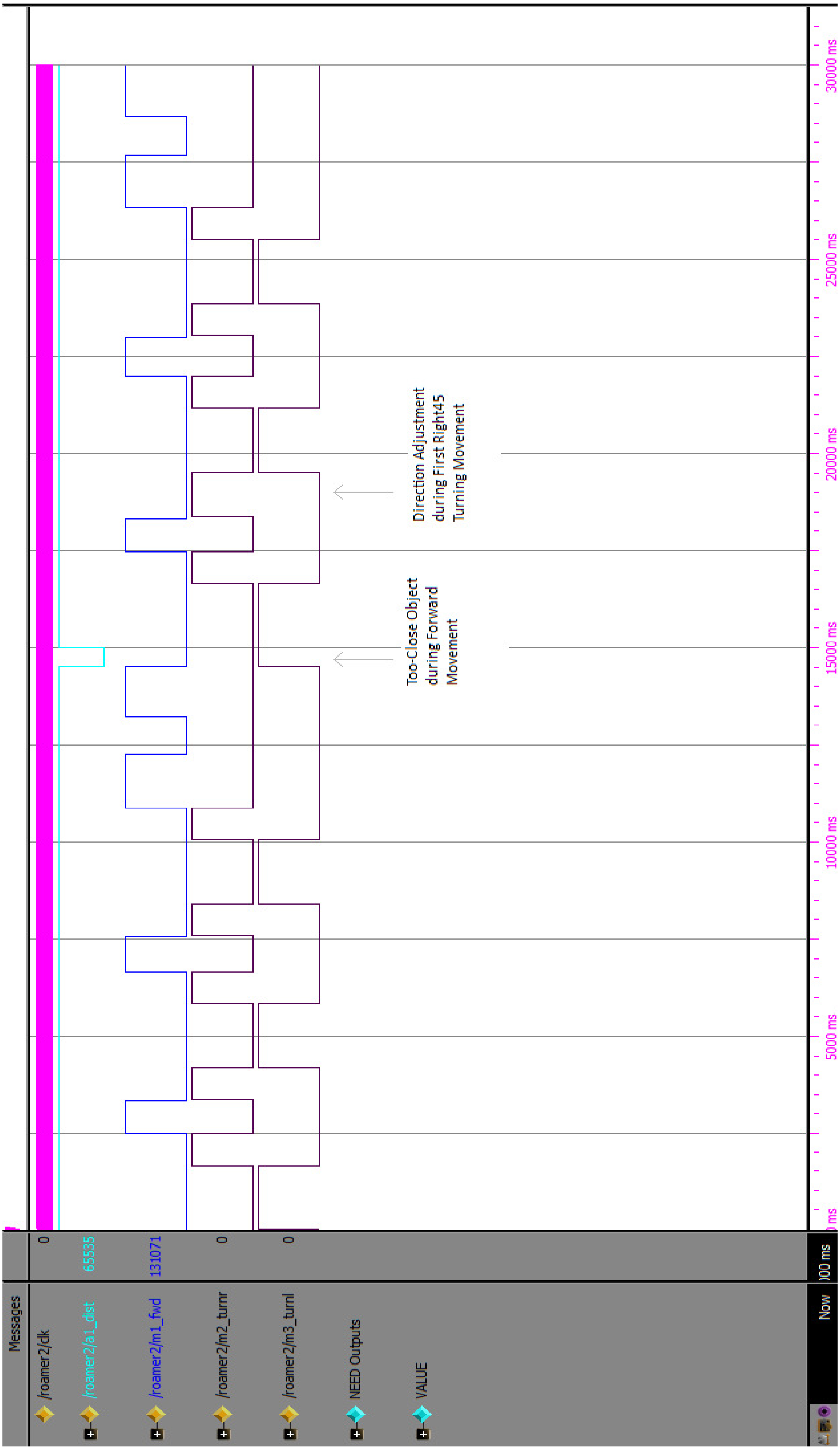


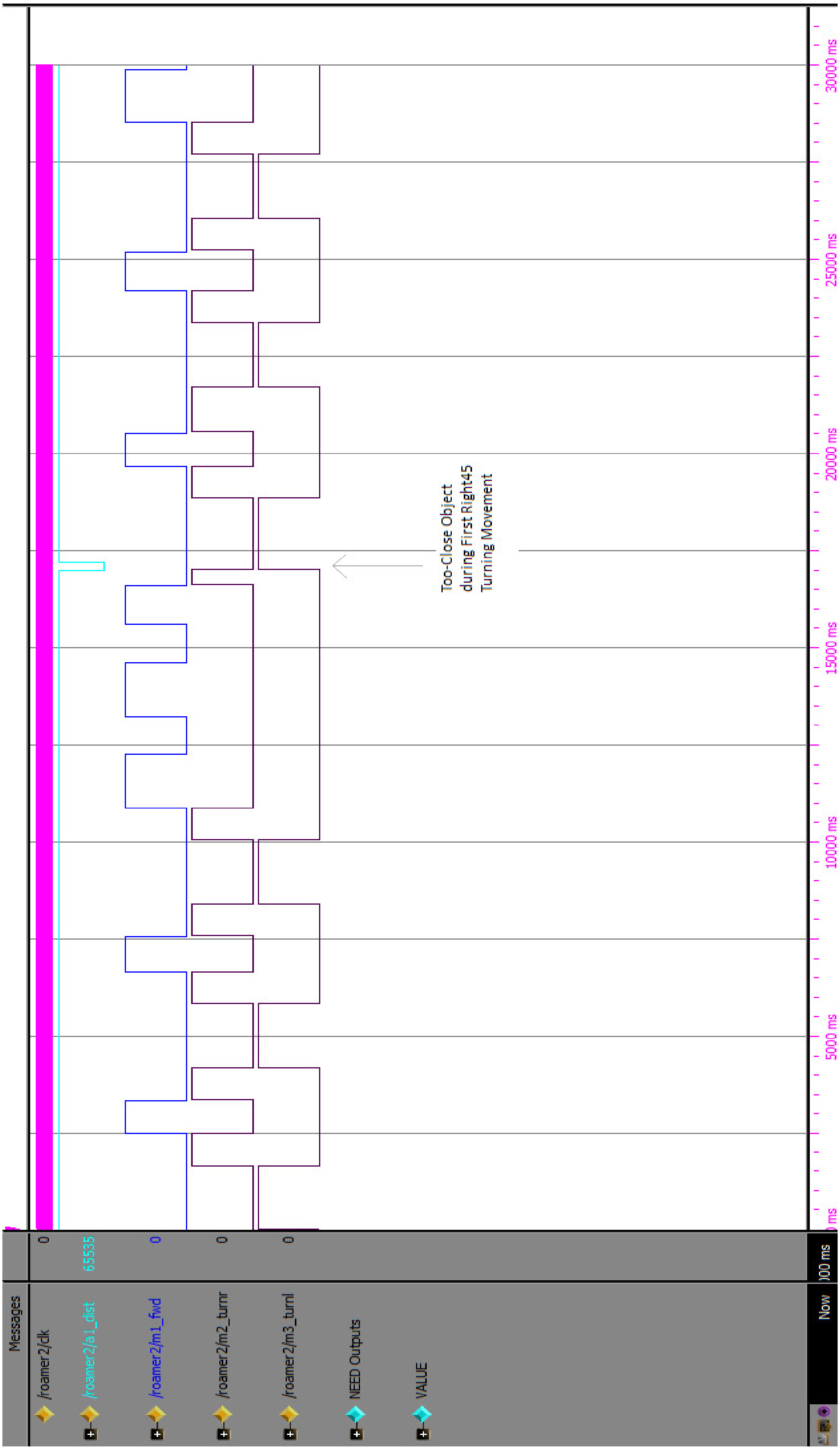
Figure 20: Complete design simulation – No too-close objects detected



Too-Close Object during Forward Movement

Direction Adjustment during First Right45 Turning Movement

Figure 21: Complete design simulation – Single too-close object during forward movement



Too-Close Object
during First Right45
Turning Movement

Figure 22: Complete design simulation – Single too-close object during first Right45 movement

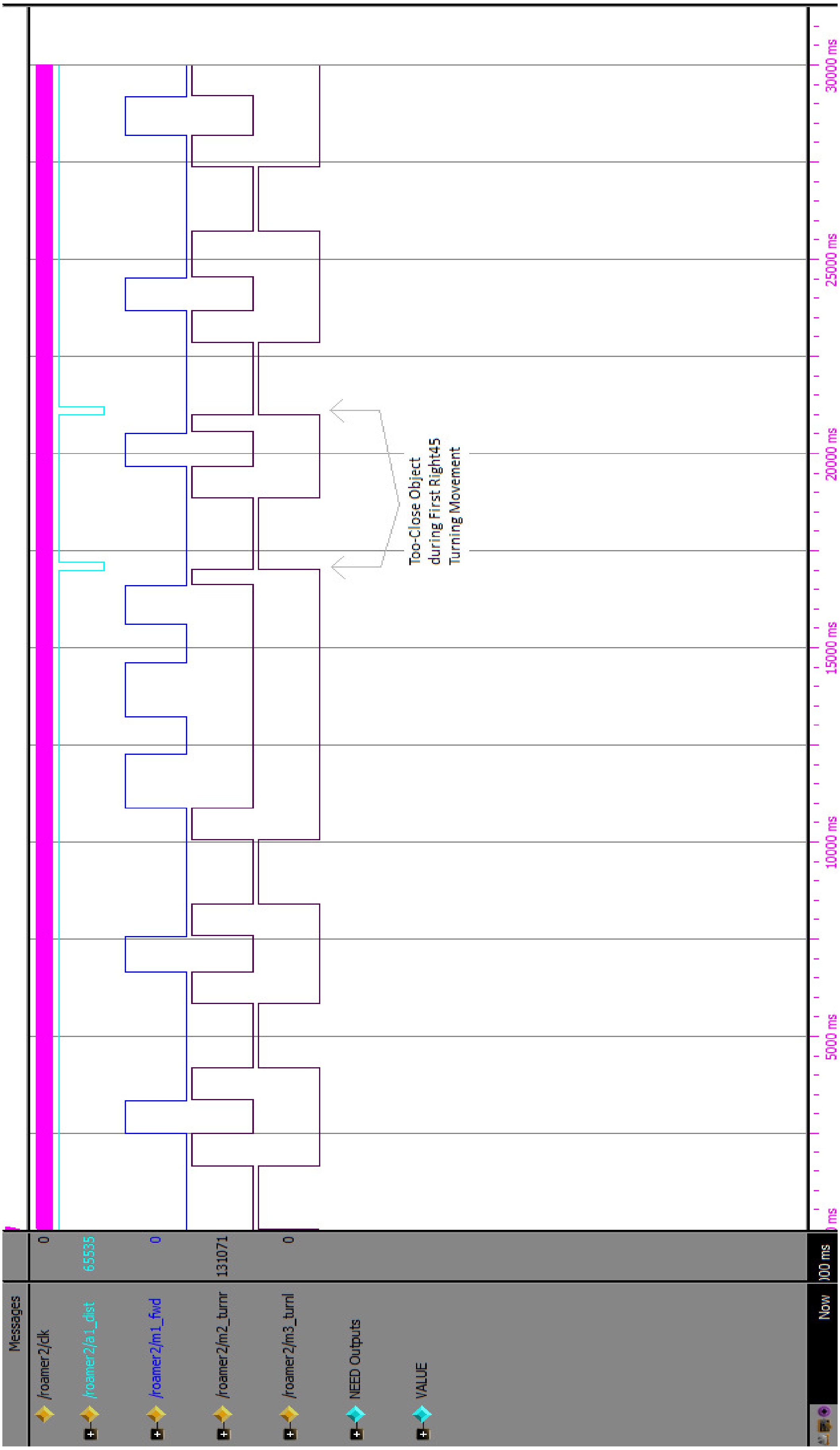


Figure 23: Complete design simulation – Too-close objects during two consecutive first Right45 movements

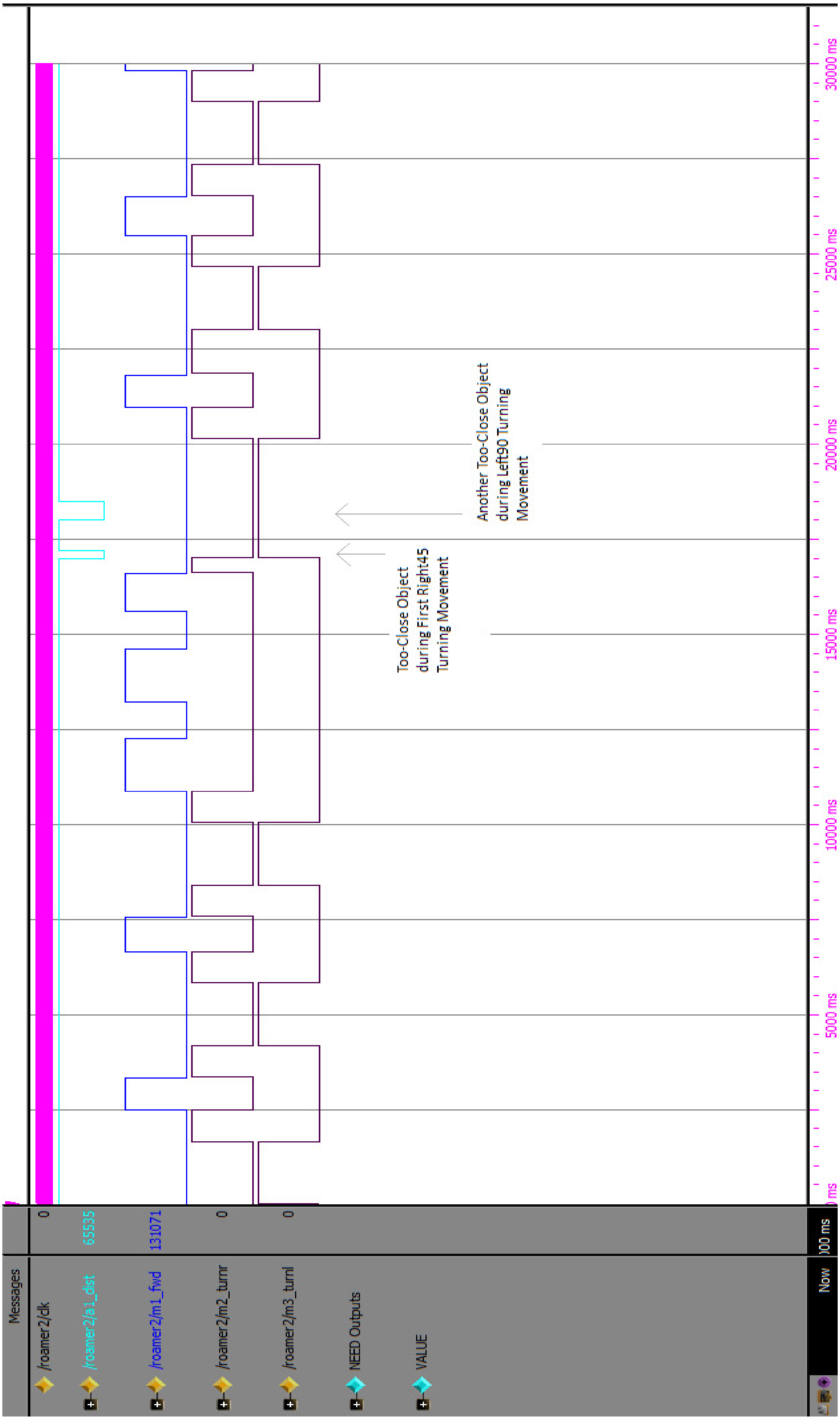


Figure 24: Complete design simulation – Too-close objects during a first Right45 movement followed by another object during the Left90 movement

3.3.3. FPGA Synthesis Results

In this section I synthesise the complete VHDL design for a specific FPGA device. The synthesis will indicate the FPGA resources consumed by the current Roamer synthetic brain. The target FPGA device is the Xilinx Spartan3A XC3S200A which has the following main features as relevant for the discussion here:

- System gates: 200K (Equivalent number of logic cells: 4,032)
- Dedicated 18x18-bit multipliers: 16

Thus, it is one of the smaller FPGAs available in the market, with the idea being to show that Roamer's current brain is still relatively small and as such make it probable that a much larger synthetic brain would still fit onto an already existing FPGA. Additionally, the VHDL model has not been optimised for a specific FPGA device or family of devices. For instance, in certain FPGA families more advanced multipliers are available that include perhaps a pre-adder, post-adder or accumulator, which are elements currently used in the VHDL model. Multipliers require a significant amount of logic resources, hence the availability of dedicated multipliers in almost all FPGAs. Thus, the currently selected FPGA might not be the most appropriate to implement Roamer's synthetic brain – that optimisation is left for future research.

Below I give two sets of synthesis results: the first shows the resources consumed by a single Need with default configuration and the second the results for the complete artificial brain. These results are obtained using the software package “Precision RTL 2008a.42”.

3.3.3.1. Need1

```
***********************************************
Device Utilization for 3S200AFT256
***********************************************
Resource                Used    Avail   Utilization
-----------------------------------------------
IOs                     148     195      75.90%
Global Buffers          1       24        4.17%
Function Generators     174     3584      4.85%
CLB Slices              87      1792      4.85%
Dffs or Latches         71      3974      1.79%
Block RAMs              0       16        0.00%
Block Multipliers       1       16        6.25%

-----------------------------------------------
```

3.3.3.2. Complete Synthetic Brain

```
***********************************************
Device Utilization for 3S200AFT256
***********************************************
Resource                Used    Avail   Utilization
-----------------------------------------------
IOs                     67      195      34.36%
Global Buffers          1       24        4.17%
Function Generators     2534    3584     70.70%
CLB Slices              1267    1792     70.70%
Dffs or Latches         408     3974     10.27%
Block RAMs              0       16        0.00%
Block Multipliers       8       16       50.00%

-----------------------------------------------
```

It is interesting to note that the Need alone require a single multiplier (as one would expect) but the complete design requires only 8 multipliers despite the fact that the design implements 14 Needs. This could be the result due to optimisation performed by Precision. Note that even for this FPGA there is some room left for variations on the current design; as an estimate, around (0,3*14) additional Needs.

3.3.4. Final Remarks

Some issues arose during the physical realisation process that are worth noting. The first concern is, based on the current design effort, to estimate what the effort would be for designing in VHDL a synthetic brain with a vast number of Needs. Hand-coding each Need and especially interconnecting them in VHDL is cumbersome and error-prone. What seems lacking is a graphical design tool that converts a graphic-based Need design automatically into VHDL. However, even for a graphical solution (as opposed to lines of text) there will be challenges of which the most obvious will be how to display all these parameters and interconnections on the screen. The design above (Figure 9) was done in two dimensions (2-D) but there is no reason why a 2-D physical device, for example the FPGA, should limit the design from being done in 3-D unless the device imposes such a constraint (for instance, two connections that represent two wires whose paths cannot physically cross, and other possible signal routing problems).

Even more of a problem might be the effort of simulating the complete design. For a single Need, simulation is still relatively simple but for the complete design, it is firstly difficult to aptly display all the Needs and their Values on the screen, and thereafter have the capability to observe and debug incorrect behaviour. A design tool that contains an effective simulation feature could be one solution.

3.4 Conclusion

In terms of design complexity and effort, the current design is already more complex than I anticipated initially. This could be because the Need model is too complex or that I have spent so much time with the design that the current version is optimised to a significant level. Optimised designs are always difficult to follow (similar to Assembly language being less readable than C language in general). On the other hand, in software a lack of optimisation leads to inefficient use of resources and yields code that executes slower. Perhaps there exists a trade-off: in order to decrease design complexity, the number of Needs will have to increase. However, I would argue that most of the complexity lies within the nature of the design – that of being multiple interconnected nodes that function in parallel. Still, one must purposefully attempt to simplify the Need model and the resulting design as much as possible.

Different from software but similar to a biological brain, there are no clear interfaces for appending or replacing sub-modules (if this is not evident at the moment, it will become clearer in the future with larger designs). One can look at this phenomenon in the following way: the current features of Roamer's behaviour, combined with the set of sensors and actuators incorporated, converge towards a balance between these elements as the design draws to completion. One can easily adjust the parameters to get a range of different behaviour (a sort of natural variation) but it is far more complex to change the structure of the design by adding or removing Needs, which will upset the balance and force one to revise possibly the complete design to reestablish a new balance with the changed structure of Needs. This is similar to what we see with DNA, where mutations are less likely to happen with those information bits that code for the structure of the organism.

With regards to conceptual thinking, it seems necessary that the designer is able to think in advance, when performing the abstract design, as to what behaviour would be caused by which Need. Whether the IMORGal design process is conceptually challenging, I cannot currently say – this will become evident when more people try to understand the concepts and design their own IMORGal. An interesting aspect of the IMORGal technology is that anyone can design an IMORGal. The IMORGal technology contains no mathematical formulas, nor does it require detailed knowledge of any field like computer science, electronic engineering, physics, psychology, biology or similar. Thus, the IMORGal technology opens up the world of building 'intelligent robots' to more people than current robotics technologies which tend to be exceptionally complex. No knowledge of VHDL or FPGAs are even required – with a simple graphical interface the everyday software user can construct his IMORGal design and at the end click on a button that will automatically convert the graphical design into an FPGA programming file that is ready for upload to the physical IMORGal. Special simulation software can be written that allows time displays similar to those shown as results for the simulations above. A generic robot base with sensors and actuators already installed can be ordered, with the aspiring synthetic brain designer left to only concern him- or herself with the design of the robot's behaviour.

That said, the greater one's dominion over digital design and other fields from which the IMORGal concept was born, the greater the ability to realise designs that are more complex and advanced than those done by the average IMORGal designer. Here as well I am not currently sure as to the difference in designs that could be realised by people coming from different academical backgrounds – this will become more evident in the future. In general, I suspect that for most readers it would be difficult to follow the design description in this section, possibly because too much information was given at once instead of explaining the design in small simple steps.

I am currently of the opinion that the current IMORGal complexity is not too complex either for understanding or implementing and designing a different IMORGal. Future designs using the currently defined Need types should be able to reach much greater complexity than Roamer and still be able to make sense when studying each element with respect to its contribution to the overall behaviour. Much of the complexity can be alleviated by creating graphical design software (mentioned above) that allows the designer to effectively deal with the increasing amount of Needs and interconnections.

Regarding scalability, there are two sides: scalability with regards to the abstract design and to the physical realisation. For the abstract design, most of what was said above for complexity is also true for scalability. That is, one can obviously just keep on adding Needs in order to get larger IMORGals with more complex behaviour, but at what point does the shear number of Needs and interconnections make the design effort become impractical? This can be dealt with partially by keeping the Needs as simple as possible and choose to have more but simpler Needs rather than fewer but more complex. At the physical layer one is limited by the size of the FPGA. However, here additional problems arise. For instance, one must take into account the number of interconnections. Each 'line' in the abstract design is actually 18 bits wide. A practical aspect is that one must consider not only the theoretical limit as defined by the biggest FPGA on the market but rather determine the biggest FPGA one has access to for implementation in an IMORGal, as well as take into account the maximum power consumption allowed by the IMORGal hardware design.

An additionally development that is already underway is to serialise the execution of the Needs in a manner similar to Time-Division Multiplexing (TDM). This refers to the following situation: the update cycle for the current design set to 100 us. That is, each Need inside the design is updated every 100 us. However, the updating sequence of the Need (the calculations that have to be made to determine the new Value of the Need) might only require perhaps 100 ns to execute. During the rest of the (100us – 0,1us), nothing happens inside the synthetic brain as far as that Need is concerned. Thus, instead of updating a 1000 Needs every 100 us using a 1000 multipliers, one could design a hardware mechanism for the FPGA whereby a 1000 Needs can still be updated in 100 us but by using only 1 multiplier. This will further allow all the synapses to become plastic, that is, changeable during operation, which will allow full adaptability throughout the synthetic brain. Thus, if we now use 2 multipliers to process a Need (one for the synapses, and the other for the Value), we still end up with 500 Needs per multiplier effectively. Take into account that the biggest FPGAs have beyond a 1000 multipliers (the Virtex XC6VSX475T has 2,016 [31]), that gives us 500,000 Needs. However, one has to remember that the greater the number of synapses that feed into a Need, the longer it will take to calculate the new Value. Ideally, the case of maximum utilisation will be when each multiplier multiplies every FPGA clock cycle. Beyond multipliers, a bigger problem will be memory. Although an FPGA already has typically a substantial amount of internal memory (the Virtex XC6VSX475T has 38,304 Kbit), one could also consider adding external memory, if the timing requirements allow it. Note that serialisation does not mean the Needs are no longer executing in parallel – it is the update cycle that counts. During that cycle, the current Values are kept constant while their new Values are being calculated. Just before the new cycle starts, all the Needs' Values are updated at the same time. Thus, parallelism is maintained.

An additional comment to add here regarding the sensor type, is that instead of using traditional analogue-based sensors that require an Analogue-to-Digital Converter to convert the analogue sensor output to a digital format as required by the FPGA, a better solution would be to use frequency-based sensors [32]. These sensors directly output a digital signal of which its frequency varies according to the measured value. Besides the frequency-based sensor output being in a format that is easier to process by the synthetic brain, each such output requires only a single I/O pin to enter into the FPGA, whereas a converted analogue signal with at least require several bits (16, in the case of feeding a Drive Input of some Need). This concept of one sensor per I/O pin is important in order to avoid the situation where a device might have enough internal resources for thousands of Needs but only a few I/O pins for receiving sensor output. Fortunately, FPGA devices in general have a lot of I/O pins. The Virtex XC6VSX475T, for example, has 840 I/O pins, with other Xilinx Virtex-6 devices reaching up to 1200 I/O pins! Obviously one would also have to take into account the signals required to drive the actuators. Similar to sensors, the ideal would be an actuator that can be driven/controlled by a frequency-based signal requiring only a single I/O pin.

There is much one can say when comparing a Need and a neuron. Obviously, they are not the same. Need types N4 and N5 share greater similarity with a neuron in that the Need/neuron fires when sufficient input surpasses a firing threshold. One core difference is that the neuron releases a single pulse (axon potential) when firing, whereas the Need outputs a constant value that is maintained until the Need's Value falls below the lower threshold. In principle, information is 'coded' or transported differently in neurons and Needs N4 and N5. Notice that N1 to N3 behave in an opposite manner to that of a neuron because those Needs trigger in the sufficient absence of input – any input applied will move the Need towards ceasing to fire.

Due to the nature of the design parameters (by which I mean every parameter directly affecting behaviour) the design of each Need does not have to be exact for the entire organism to function. A substantial part of the design phase is that of trial and error, of experimentation that includes adjusting the design parameters until the desired behaviour is achieved. Even more, it is here where evolutionary methods could be used effectively. One only has to specify a range for each parameter (or those parameters one is not certain of) and then iteratively run through a process of randomly changing each of these parameters, visually see which behaviour is the one I am interested in, and then save and use those parameters in future. If I am not happy, I keep on applying the mutation principle to those IMORGal that behave 'better' to me. Specifically for this case, I am is therefore the fitness function for the evolutionary algorithm.

4. THE FUTURE

What are the next steps in the evolution of the IMORGal technology? Generally speaking, I am convinced that the details of the technology (especially the Need implementation) will change over time. Until the foundation (mentioned at the start of this document) is reached, I will remain slightly skeptical of the validity of the IMORGal theory and the technology it proposes as a solution. In terms of specific objectives, I suggest the following approach:

- The next immediate step is to develop effective design tools, as discussed in the previous section.
- In parallel, the serialisation of the Need updating must be completed since the current FPGA is already, roughly speaking, at half its capacity. Including synaptic plasticity will immediately consume the remainder of the FPGA's resources. Therefore, to enable expanding the current design, the FPGA's resource usage must be optimised first via the serialisation mechanism.
- The physical Roamer was programmed and given an initial run to see how its behaviour compared with the design (details about the physical experiment are not included here because no results are ready for publication at the moment). However, immediately the following aspects were noted:
  - Obviously, the first issues have to do with the specifics of the sensor and wheels. For instance, the sensor's performance is affected by its positioning on the robot. In addition, no two sensors have the exact same performance. The weight distribution across Roamer can have the effect that one wheel has to work harder than the other in order to maintain a straight line during forward movement. These are issues that need to be dealt with by Roamer's adaptability (which is not yet present in the current design).
  - There are factors that influence the performance of Roamer in terms of not touching any object. For instance, the turn angle during the safe zone check might not be wide enough. It could be that Roamer reacts to slowly to the detection of a Too-Close object – Roamer needs to stop immediately, whether turning of moving forward. These issues also fall under elements that adaptability need to take care of. However, these issues operate on a longer time scale. For instance, over a period Roamer can learn whether it keeps on touching objects at regular intervals and thus needs to, for example, increase the surface of the safe zone area. These parameters could also be improved using evolutionary means.
  - To add adaptability implies adding more sensors that will enable the necessary information to be extracted from the environment to serve as feedback into the synthetic brain, and which is required for the adaptation process

  Thus, in terms of expanding the current synthetic brain design for Roamer, the next features to be added will be related to adaptability. The main feature to be added will be:

  - At the moment there is an a priori wheel speed set for each wheel when Roamer moves forward or turns. This is done in a blind manner, that is, Roamer sets the value but does not know if the wheel is actually turning. It has no sensory feedback to perceive, for instance, a situation where one motor is broken (leading to the wheel never turning), which leads to all forward movement basically resulting in turning in circles. Thus, the first ability should be to perceive what is happening to each wheel and use this information to ensure that, during forward movement, both wheels turn at the same speed (ensuring straight line movement). If a wheel turns too slowly than the set value, the speed applied to that wheel should be increased – if turning to fast, then the speed should be decreased. For turning, a

  similar effect can be implemented to ensure that the turning angle is according to the design (or set value).
  - A further enhancement to Roamer's adaptability regarding the speed of each wheel, is to avoid programming the desired speed value of each wheel a priori, but rather give Roamer the ability to determine the speed by itself. This will require that much more information be extracted from the environment, that is, more sensors of possibly different types that provide Roamer with the necessary information.
  - Instead of programming the safe zone distance a priori, rather give Roamer the ability to determine by itself what a valid safe zone distance is that will ensure no contact with any object within the current environment. There could be several issues that one has to deal with. I mention one: In order to learn what a valid safe zone distance is, Roamer will most likely have to touch objects several times. It could be that touching of any object is never allowed (for example, a situation where, functioning between humans, 'touching' a human could lead to injury). One could consider perhaps a special training ground for Roamer to establish its safe zone distance value.
- All information that can be extracted from the current sensor, should be extracted first before moving on to using different sensor configurations (for example, two distance sensors of the current type). Further information that could be extracted are:
  - Static versus dynamic (moving) objects. That is, if an object is detected, it could be that the object is static and Roamer was turning or moving forward, or it could be that the object is dynamic. To determine the difference, Roamer will have to able to sense its own movements. If Roamer is not moving and there is a variation in the sensor measurement, it is likely that the object is dynamic.
  - Object identification using a frequency signature. For example, I could place different fans around the space that turn at different velocities. The fan blade movement will cause a fluctuation in the value measured by the distance sensor. This could be used by Roamer to identify the fan. One can build into the design, for example, that a certain fan is a negative element, causing Roamer to temporarily increase the surface of its safe zone search. Another fan could have a positive connotation to indicate that the immediate surroundings have low risk in terms of touching objects. Additionally, the same fan can use different velocities to indicate its state: for example, high velocities indicate safe roaming conditions and low velocities unsafe roaming conditions
- In terms of actuator complexity, it would be best to keep the motor functions of Roamer as simple as possible for as long as possible. Rather concentrate on:
  - implementing more sensors of the same type,
  - implementing different sensor types that have increased complexity (that is, they allow more information to be extracted from the environment),
  - implementing a combination of different sensors in the same design (this will enable multimodal perception).

One potentially serious issue is that of hardware, that is, having at one's disposal a compact robotic mobile platform that allows flexible sensor addition. Alternatively, one could consider using existing robotic devices that were designed for different projects. A retrofitting approach could allow the addition of a small synthetic brain module (which basically just contains an FPGA and I/O connectors) to the robotic device, interfacing directly or indirectly (via a microprocessor) with the device's sensors and actuators. Such a synthetic brain module can be available as well to those who wish to implement a synthetic brain themselves. The question of hardware will thus have to be addressed in the short term.

5. BIBLIOGRAPHY

[1] Japan Trends. Paro Robot Seal – Healing and companionship pet. Web Page.
[Visited: 2009-12-09]. Available at:
http://www.japantrends.com/paro-robotic-seal-healing-and-companionship-pet/

[2] Auke Ijspeert - Salamander robot swims and walks. Web page.
[Visited: 2009-12-09]. Available at:
http://lis.epfl.ch/index.html?content=resources/podcast/2007/03/auke-ijspeert-salamander-robot-swims.html

[3] Robots: From Animals to Automation - Festo. Web page.
[Visited: 2009-12-09]. Available at:
http://www.robotspodcast.com/podcast/2009/07/robots-from-animals-to-automation.html

[4] Barbara Webb - Biorobotics. Web page.
[Visited: 2009-12-09]. Available at:
http://lis.epfl.ch/index.html?content=resources/podcast/2006/10/barbara-webb-biorobotics.html

[5] Technology Review. Rat-Brained Robot. Web page.
[Visited: 2009-12-09]. Available at:
http://www.technologyreview.com/Infotech/13047/?a=f

[6] Talking Robots - The Podcast on Robotics and A.I. Web page.
[Visited: 2009-12-09]. Available at:
http://lis.epfl.ch/index.html?content=resources/podcast/

[7] Robots Podcast & Community. Web page.
[Visited: 2009-12-09]. Available at:
http://www.robotspodcast.com/index.php

[8] I. Harvey, Robotics: Philosophy of Mind using a Screwdriver, Evolutionary Robotics: From Intelligent Robots to Artificial Life, Vol. III, T. Gomi (ed), AAI Books, Ontario, Canada, 2000. pp. 207-230. ISBN 0-9698872-3-X.

[9] Iwasaki Lab. Neural Network of the Nematode C. elegans. Web page.
[Visited: 2009-12-09]. Available at:
http://ims.dse.ibaraki.ac.jp/research/C_elegans_en.html

[10a] Brooks, R. A., "New Approaches to Robotics", Science (253), September 1991, pp. 1227–1232.

[10b] Edge. "The Deep Question" A Talk With Rodney Brooks. Web page.
[Visited: 2009-12-09]. Available at:
http://www.edge.org/3rd_culture/brooks/brooks_p1.html

[11] Robotics Trends. Web page.
[Visited: 2009-12-09]. Available at:
http://www.roboticstrends.com/

[12] SRI Consulting Business Intelligence, Disruptive Technologies APPENDIX E Background: Service Robotics, Global Trends 2025

[13] iREX 2009 - International Robot Exhibition 2009. Web page.
[Visited: 2009-12-09]. Available at:
http://www.nikkan.co.jp/eve/irex/english/

[14] International Federation of Robotics. Service Robot. Web page.
[Visited: 2009-12-09]. Available at:
http://www.ifr.org/service-robots/

[15] Aldebaran Robotics. Nao's World. Web page.
[Visited: 2009-12-09]. Available at:
http://www.aldebaran-robotics.com/eng/Nao.php

[16] Robotis. Business Area. Web page.
[Visited: 2009-12-09]. Available at:
http://www.robotis.com/zbxe/business_en

[17] Precise Path Robotics. Introducing The RG3 Robotic Greens Mower. Web page.
[Visited: 2009-12-09]. Available at:
http://www.precisepath.com/

[18] "The Strategic Research Agenda for Robotics in Europe", European Robotics Technology Platform, July 2009

[19] Floreano, D. and Urzelai, J. (2000) Evolutionary Robotics: The Next Generation. in Evolutionary Robotics III, Ontario (Canada) : AAI Books, 2000.

[20] Floreano, D. and Mondada, F. (1996) Evolution of Plastic Neurocontrollers for Situated Agents. From Animals to Animats 4.

[21] Laboratory of Intelligent Systems (LIS). EPFL. Web page.
[Visited: 2010-01-06]. Available at:
http://lis.epfl.ch/

[22] Brooks, R. A., "How To Build Complete Creatures Rather Than Isolated Cognitive Simulators", Architectures for Intelligence, K. VanLehn (ed), Erlbaum, Hillsdale, NJ, Fall 1989, pp. 225–239.

[23] Keijser, F. A., "Making decisions does not suffice for minimal cognition", (2003, Adaptive Behavior, 11(4) , 266-269)

[24] Krichmar, J.L. (2008). The Neuromodulatory System - A Framework for Survival and Adaptive Behavior in a Challenging World. Adaptive Behavior, 16, 385-399.

[25] Scholarpedia. Neurorobotics. Web page.
[Visited: 2009-12-09]. Available at:
http://www.scholarpedia.org/article/Neurorobotics

[26] The NeuroSciences Institute. Brain-Based-Devices (BBD). Web page.
[Visited: 2009-12-09]. Available at:
http://www.nsi.edu/index.php?page=ii_brain-based-devices_bbd

[27] J.L. Krichmar, A.K. Seth, D.A. Nitz, J.G. Fleischer, G.M. Edelman (2005) Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions. Neuroinformatics, 3(3), pp. 197-222

[28] Edelman, G. M., November 2007. Learning in and from brain-based devices. Science 318 (5853), 1103-1105.

[29] A.K. Seth, O. Sporns, J.L. Krichmar (2005) Neurorobotic models in neuroscience and neuroinformatics. Neuroinformatics 3(3), pp. 167-170

[30] Scholarpedia. Action Selection. Web page.
[Visited: 2009-12-09]. Available at:
http://www.scholarpedia.org/article/Action_selection

[31] Xilinx. Virtex-6 Family Overview. DS150 (v2.1) November 6, 2009

[32] IFSA - International Frequency Sensor Association. Web page.
[Visited: 2009-12-09]. Available at:
http://www.sensorsportal.com/

[33] Werger, B. B. 1999. Cooperation without deliberation: a minimal behavior-based approach to multi-robot teams. Artif. Intell. 110, 2 (Jun. 1999), 293-320.

[34] Antidiuretic Hormone (Vasopressin). Colorado State University. Web page.
[Visited: 2010-01-06]. Available at:
http://www.vivo.colostate.edu/hbooks/pathphys/endocrine/hypopit/adh.html