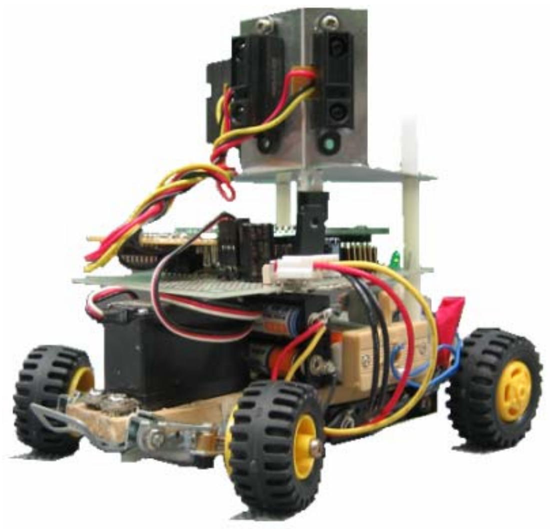
DEMAMECH 2005-2006 exchange student report

Personal Data

Name:	Maarten Vaandrager
Email:	vaandrager@gmail.com
Home Institute:	Delft University of Technology
Faculty:	Faculty 3ME,
Department:	DCSC, Delft Center for Systems and Control
Address:	Mekelweg 2 ,2628 CD, Delft, The Netherlands
Supervisor:	Prof. R.Babuska
Host Institute:	Hokkaido University
Faculty:	Information and Systems Science
Department:	Hybrid systems laboratory
Address:	Kita 14 Nishi 3 Kita-ku, Sapporo, Japan
Supervisor:	Prof. H. Igarashi



強化学習

北海道大学 HOKKAIDO UNIVERSITY

LABORATORY OF HYBRID SYSTEMS

DCSC

TU Delft

Executive Summary

After having said everybody goodbye I departed on the 3rd of February to many yet unknown friends and new experiences. After a three day stop in Hong Kong I finally landed at Chitose airport and was picked up by my first overseas friend, a Brazilian PhD student from the laboratory. He knew exactly what I needed as a newcomer to this wonderful but sometimes alien land. The first days he helped me with registrations and finding my way to and fro the basic necessities of life (dormitory, laboratory, convienentstore). I can't stress enough how nice it is to have someone accompany you through the first days exploring your new environment.

Having spent the first few days exploring, next was getting some real work done. My research topic was called "Reinforcement Learning for a FPGA based mobile car". Since I still had only a superficial understanding of what Reinforcement Learning was and how you can use a FPGA (Field programmable gate array; a reconfigurable chip for implementing logic circuits) for this purpose the best start was reading huge piles of books, papers and extracting everything that could possibly be relevant or useful for my application of the theory. My professor also helped in this by dumping some frighteningly big books on my table. This period of reading I have experienced as very pleasant since the subject really caught my interest and I learned a lot not by following some dull course but actively researching this subject with a clear objective in mind. In this period I also discovered many cultural curiosities like the following examples:

Laboratory

The Laboratory is a very very cozy place. It doesn't seem to be ruled so much by the professor whose lab it really is but more by the students who spend so much time in it. Because of this the lab really is a nice place to hang around. You can crawl under you desk to take a long sleep or just put your face on you keyboard to take a short nap just like everybody else is doing. And it is even possible, as one lab member has shown for several months, to live there and only leave to take a shower and do groceries. (the buildings are open 24/7!) People work for a long time (years) on the same subject so that they really have thorough understanding of the specific theory and try to make a practical application. This results in more real research done by mere students and makes university a lot more fun and exciting than the huge piles of 'basic' theory Dutch students have to absorb but never really put to practice. And since there is the possibility for companies to sponsor research in the university you don't have 'independent scientific research' as in we Holland but there seems to be a lot more hard cash available for student projects which again makes university more fun and exciting. The biggest problem for me in this laboratory was the language barrier. This really felt as a barrier since conversations never got beyond 'nice weather', 'I'm sleepy', 'I like this'. And although I had some good friends there my best friends were still other English speaking international students.

Unemployment

Every-body is employed in Japan. Unemployment is virtually non-existent. And on top of this everybody is really working to their very best. This makes Japan a really productive country. The only drawback is that many jobs seem a bit useless to me. Especially in Sapporo there were many people 'directing traffic' by waving a stick, saying welcome to entering customers while the other employee is there to help them, people rearranging bikes in a neat row or waving a sign with ads on it, etc. Though most Japanese say these jobs are essential, I can't help but be skeptical.

Friendliness and Indirectness.

Japanese are very polite. Every one, every time. Even a mean looking punk will be glad to help you in any way he can. The politeness and friendliness is a very nice thing and sometimes difficult not to take a bit advantage of. I guess the whole world should take Japan as an example for a society where friendliness and politeness rule the streets. But also here there is a drawback which is that sometimes you don't need politeness but clarity. And this is sometimes hard to find between the many friendly smiling faces. A little bit… this means 'no'. eehm… means 'no way!'. I hadn't noticed in the laboratory when I did something that wasn't appreciated until some days later there was an English sign saying "please do / don't …" One thing I definitely would have liked is some more honest criticism on my work.

Groups.

You're either in or out. If you are in, I doesn't matter who you are, everybody accepts you as you are. But if you are not in, it could sometimes even be, for a person from a certain group, rude to the rest of his group to talk to you. Because of this the group is a very important thing for Japanese. The laboratory is a group and there is almost no communication between students of different laboratories. The company they will work for is a group for which they will probably keep working for the rest of their life. Also because of this group feeling the company will never fire them. This might be the reason for people working in 'seemingly useless' jobs just for the sake of employment and not having to fire people.

Japanes / English

Japanese can't speak English. Of course this is not true since there are many English speaking Japanese especially in the Tokyo and Osaka areas but at least here in Sapporo there are so few that it a dare say Japanese can't speak English. It is important to realize however that they keep learning Japanese language until the end of high school and than still know only part of the kanji's. Many people have a dictionary with them to look up kanji's they don't yet know. And while they are still learning their own language they're also supposed to learn English which is so immensely different from Japanese. I had never realized how different two languages can be. The Japanese 'alphabet'; the 'Kana' consists of about 46 syllables. The amount of syllables that can be made from the alphabet is of course less than 26x26=676 but it is still a lot more than the 46 Kana. Therefore there are many English words they cannot pronounce correctly or hear the difference since they are not used to these sounds. So maybe for a foreigner Japanese language looks complicated, I'm sure English looks even more complicated to a Japanese person.

Travel schedule

3rd of February	Flight from Schiphol(Amsterdam) to Hong Kong
7th of February	Flight from Hong Kong to Chitose(Sapporo)
4th of August	Boat from Toyama to Vladivostok
6th of August	Transsiberian train to Moscow
26th of August	Arrival in Delft, Netherlands

Research

The goal of the project was to control a small robotic car by use of a Reinforcement Learning algorithm implemented into a FPGA. The robot's task was simply to drive around a course without bumping into the walls. This sounds simpler than it is because at the start the robot does not have any information about the relationship between its sensor output, environment and the consequences of its actions. The Reinforcement learning should make the robot able to understand these consequences and take the appropriate action depending on it's perception of the environment by its sensors

First I have tried to read about and understand the Mechanism of Reinforcement Learning and decide which implementation was most suited for this problem. After this I have tested my understanding and ideas for implementation by simulating the problem using matlab and building controllers to test them in this simulation. Finally I learned the VHDL programming language and implemented the design into the FPGA.

Theory

Reinforcement learning is learning what to do so as to maximize a numerical reward signal. The robot should be able to do this by using a value function which represents the expected future reward and using the difference between expected reward en received reward as an error signal for learning the correct value function and policy. There are many different approaches to this but here we will only discuss the most popular which is the Temporal Difference Method:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t),$$

Here the error is δ_t, s_t is the state in a gives time step t, the received reward in the next step is r_{t+1}, the expected future reward from the next step on is $V(s_{t+1})$, the expected reward in step t is $V(s_t)$ and γ is the discount of future predictions. In this formula the $r_{t+1} + \gamma V(s_{t+1})$ part is the *perceived* cumulative reward after step s_t and the $V(s_t)$ is the *predicted* cumulative reward after step s_t. By taking the difference between the predicted and perceived reward you get the temporal difference error which is used modify the function V(s) into the right direction. At the same time the temporal difference error can be used to modify the policy function which determines the best action in a given state.

So what we have are two functions which have to give a value for a given state. One gives a Value for the expected cumulative reward and the other gives a value for the action to be undertaken. I looked at two approaches for structuring these functions. One is the Actor Critic and the other is known as Q-Learning. In the Actor Critic the Value function is called the Critic and the action function (also called the policy) is called the Critic.

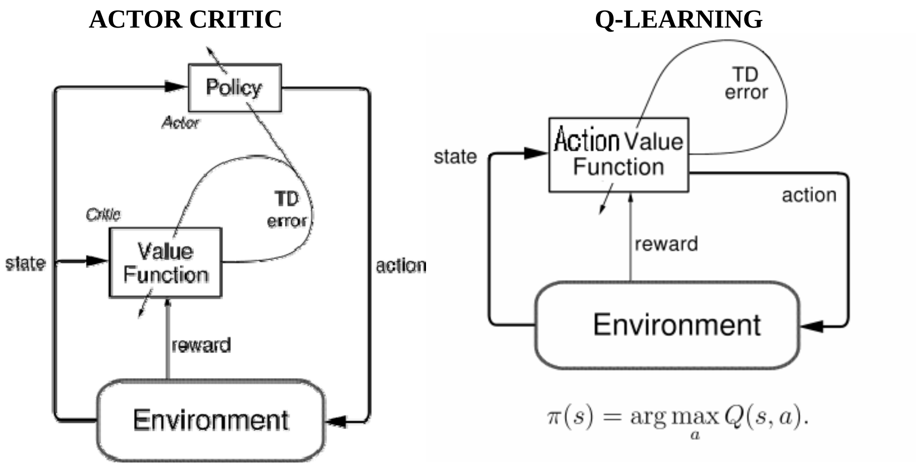


Figure 1. Left: the actor critic architecture. Right: the Q-learning architecture

In Q-learning these two functions are combined into one function $Q(s,a)$ Which gives an expected cumulative reward for every possible action and the action taken is simply the one with the highest expected cum. reward. The disadvantage however is that a lot of computation has to be done if there are many possible actions to choose from since the function has to be evaluated for every one of them to be able to choose the best or use optimization techniques to make selection more efficient.

Simulation

To be able to test the theory and to determine the efficiency and performance of these algorithms I created a simulation using Matlab which put the robot in a simple square donut and gave it three sensors to determine its place in the environment. The reading of the sensors was passed on to the algorithms as also a reward signal being simply -1 in case the robot touched a wall and 0 everywhere else. This should be enough to for the algorithm to determine the best behavior given any state s to maximize its reward over time.

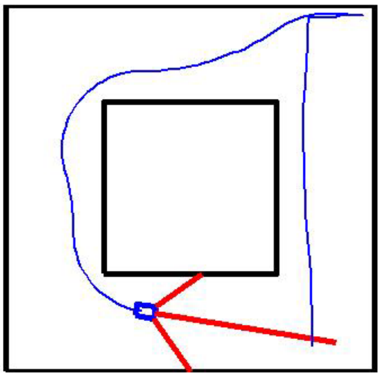


Figure 2. The simulated environment showing the walls in black, the robot in blue(including a part of its path) and sensors in red

There were several parameters which were to be optimized for example the learning rate alpha and discount rate gamma. Alpha is the rate of learning. If it is too small, learning will be slow but a too large alpha makes the algorithm unstable. Discount rate gamma determines how future rewards will be accounted for. A discount rate of 0 will yield in this case an ever growing Value function and a discount rate of 1 means the Value function is neglected and the functions are only updated by the immediate reward thus not using its knowledge about expected reward.

Optimization

The optimization was done using two methods; gridsearch and genetic algorithms. The gridsearch simply simulates the robot 10 runs of driving 250 meter and averages the received reward over all the runs. This is done for every point in a grid spanned by 20 values of gamma and alpha each. The highest point on the surface created by plotting the averaged rewards is at the optimal combination of gamma and alpha (see fig. 3 left). This method searches in 2 dimensional space and took a very long time to calculate for all different architectures.

A more efficient optimization method I used is a genetic algorithm which creates a random set of individuals with the parameters as genome. These individuals are then evaluated in the simulation and according to their fitness (in this case the accumulated reward) selected for the next generation. Of this next generation part is crossed over with each other (swapping of parameters) and part is mutated (randomly modifying parameters). After a couple of generations the population would converge to the fittest (optimal) set of parameters (see fig. 3 right).

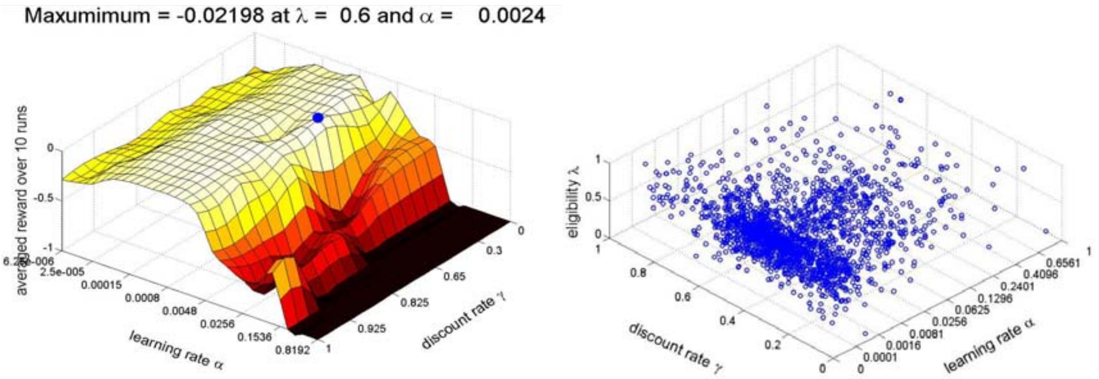


Figure 3. Left: using gridsearch to optimize parameters (the optimal combination is at the blue dot) Right: using genetic algorithm to optimize parameters. (The optimal combination is at the center of the dot cloud)

Having optimized the parameters for several different architectures of Reinforcement Learning it was time to determine which architecture was best suited for the given problem. this was done by running the simulation for every architecture several hundred times and plotting the averaged rewards as a function of distance covered by the robot (see fig. 4)

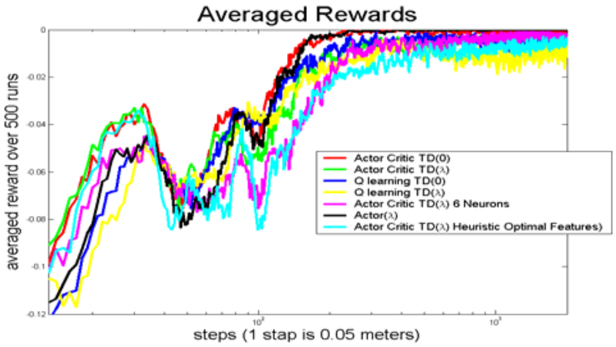
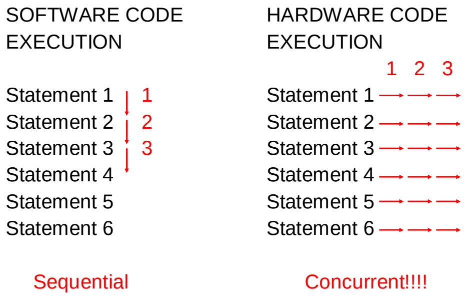


Figure 4. The averaged rewards of several different architectures of the Reinforcement Learning algorithm. The best being the one with the highest overall reward.

In the above plot its not overly clear which architecture is best but the Actor Critic is performing just a bit better than the actor critic and since the actor critic architecture is simpler to program into the FPGA this is the most obvious choice for implementation.

Implementation

The chosen design is to be implemented into the FPGA. The FPGA is a programmable chip but if you think programmable means so much like writing some software then programming this thing will be something of a shock since it involves defining every bit of logic that has to be done. This chip doesn't have memory for storing variables nor any easy debugging tools. This makes the conversion from a software algorithm to a hardware algorithm not an easy task. But the FPGA also has some distinct advantages over software. The most important is the fact that every line in hardware code is executed concurrently (parallel) instead of sequentially as in software. This allows for massive throughput of data in case of simple calculations as fig. 5 shows. Another advantage is that hardware is purely deterministic in time which means that latency of the system is virtually 0.



SOFTWARE CODE EXECUTION

Statement 1 1
Statement 2 2
Statement 3 3
Statement 4
Statement 5
Statement 6

Sequential

HARDWARE CODE EXECUTION

1 2 3
Statement 1 → → →
Statement 2 → → →
Statement 3 → → →
Statement 4 → → →
Statement 5 → → →
Statement 6 → → →

Concurrent!!!!

Figure 5. three clock cycles in software and hardware. The red arrows show the execution of a line and it shows the hardware code executes every line every clock cycle and therefore has a much bigger throughput of data than software code.

The components of the robot are partly on a small board and partly programmed into the FPGA. The major components and their functions are listed below:

Board components

1. FPGA — Programmable chip used to implement the FPGAcomponents
2. AD converter — Used converting the sensor voltage to digital values.
3. Clock — 1 MHz clock signal for the FPGA and AD converter.
4. In/out buffers — Used for stepping of voltage between components.
5. Jtag connector — Connector to upload bit stream from computer to FPGA.
6. Servo motor — Used to steer the robot.
7. Volt. Regulators — Used for stabilizing and reducing voltage.

FPGA components

8. COUNTER — Keeps track of time and used for PWMsignal creation and timings.
9. Din shift register — Used for selecting analog channel to be converted to digital.
10. Dout shift register — Used for storing the serial digital measurements into memory.
11. PWM Signal — Pulse-Width-Modulated Signal used for actuating the servo motor.
12. Timings — Used for triggering processes in the other components.
13. ACTOR CRITIC — Used for learning and determining steering angle.
14. RAM Block — Memory used for storing learned functions and previous states.

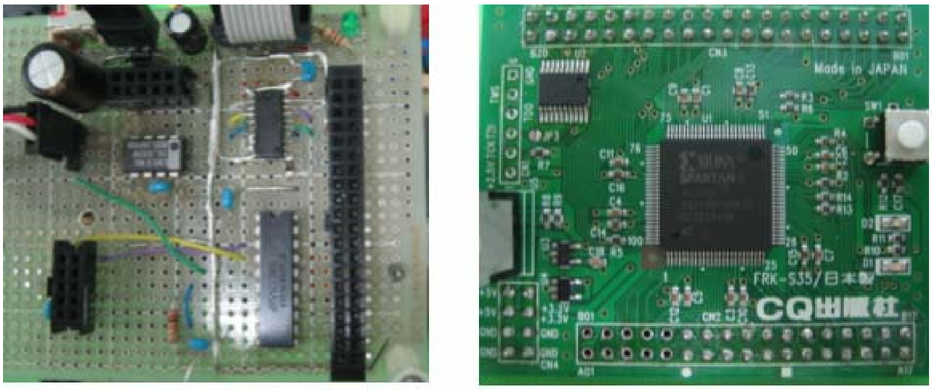


Figure 6. Left: the Board components Right: the FPGA

Conclusion

The simulation and optimization worked well but proofed one important aspect of the problem. The optimization showed that the algorithm even worked fine with a value of 0 for gamma. This means the Value function is not a necessity and the appropriate behavior can be learned by using the immediate reward only. It comes down to the fact that the task for the robot is simple enough for a feedback controller to accomplish and there is no need for Reinforcement Learning. Only when the task is more difficult than simply minimizing a well definable error (in this case staying in the middle of two walls) is there a real purpose of applying a more complex algorithm as Reinforcement Learning. Second and more important is the difficulty of implementing the algorithm into the chip. The chip is not so easily debugged as for example software code. Especially a Learning Algorithm with somewhat unpredictable and chaotic behavior makes debugging a very difficult task.

Exchange student life

Below I will give a short fictional description of my everyday life at Hokkaido University. This is not a real day during my stay but as I remember it to be.

Every weekday I have to get up at 8 o'clock since the language course starts at a quarter to 9. After a shower in our dirty, mouldy and smelly shower I skip breakfast since other exchange students have stolen my spoons and in the kitchen everything is dirty anyway. At least I am in time to have a cup of coffee before class starts. After the first one and a half hour class of grammar I get some sando's (sandwiches) at the co-op and get ready for the next one and a half hour class of Kanji writing.

At twelve I head for the Laboratory where my computer has just stopped a 12 hour simulation which I started last evening before I went home. I feel guilty waking up Hirahatake san who has spent the night sleeping under his desk and *was* still asleep before I entered. The results look good but not exactly what I had hoped for and I spend some hours plotting and rewriting the program to do the next simulation. I want to be ready before half past 6 because it's Friday and we plan to head to Susukino (the nightlife district) with some other exchange students. During a coffee break I try out some of my freshly learned verbs and nouns but the conversation is still rusty. Though a bit frustrated about my own slow progress in learning Japanese, I do notice that the English proficiency in my lab has gone up during the time of my visit. Some lab members have definitely become more fluent than some months ago.

Then, me and my best exchange student friends head for the 'sushi place'. We have been looking for other sushi restaurants but we have one preferred sushi restaurant in the middle of Susukino because it is cheapest and most delicious at the same time. After this we get a call from 'ze frenzchman' that he and some others are going karaoke and then to a new nightclub and probably ending up in the gaijinbar 'booty'. We decide to skip this time since we want to stay fit for the biking trip we are planning on Sunday and I also want to join the Sumo tournament in our dormitory again. I still want to beat that big fat guy who threw me out last time.

We head back to the Laboratory which has now become more crowded than during the day. There is some Japanese pop music and they are preparing the beamer to watch a football match after midnight. I spend some time browsing the internet and printing some papers before I go home to read them. When I leave the building at twelve it has started snowing again. Sometimes I start wondering if the winter ever comes to and end in Sapporo…

北海道大学 HOKKAIDO UNIVERSITY · LABORATORY OF HYBRID SYSTEMS · DCSC · TU Delft

Summary

For six months I lived in Sapporo. In these six months I have worked on a research project and followed a Japanese language course. The goals I set for myself during my stay are: Learn about Reinforcement Learning theory, learn the basics about FPGA’s and what it takes to program them and what their capabilities and limits are, learn basic Japanese language and communication skills. Learn to use Matlab for extensive simulation and testing, implement a fuzzy controller as opposed to the Reinforcement learning solution.

All of the above goals I reached in a certain extend and I’m very happy with the results. Working in a different environment was stimulating and made me really want to achieve some goals in the limited time that was given to me. The laboratory really felt like ‘home’ and the dormitory was dirty but nonetheless an enjoyable place to live. The only thing that lacked was some honest criticism and guidance on my work. Looking back at the period it seems as one of the most productive periods of my studies and wish I could do it more often and for longer periods. I wish everybody the same experience and am very grateful for having had the opportunity myself.