5CCS2RGP
Robotics Group Project, 2012/13
Individual Project Report

Last Name:	Vitek
First Name:	Tomas
Student Number:	1156798

I declare that I understand the nature of plagiarism as defined in the Undergraduate Handbook and that the content of this report submission is entirely my own work.

Contents

Abstract

This report evaluates the Robotics Group Project of designing and building a robot and developing a control software for it. It is intended to inform the reader about the different project stages, the team's approach to those stages and individual contributions to the project by the author. It also includes analysis of the final solutions. This document is a part of a group project submission.

1 Introduction

For this robotics group project we were given two tasks, each for one term. During the first term we were given a task to build and program a line following robot, which then at the end of the term was supposed to race against a robot made by the second group working on the same problem.

During the second term we were given a task to build and program a ball collecting robot. Similarly to the first term task, at the end of the second term there was a competition where our robot run against a robot made by the second group working on the problem with the aim of collecting more ball than the opponent.

Both robots we supposed to be build using LEGO *Mindstorms NXT [1, 2]* kits and programed in *BricX Command Center IDE* [3] using *Not eXactly C (NXC)* language [4]. The robots are build by connecting provided motors and various sensor to an intelligent LEGO brick, which can be programmed using *NXC* to utilise those motors and sensor. All remaining mechanics can be build using LEGO *Technic*.

Even though we had been attending *Adaptive and Robotics Systems* module (taught by *Dr Thrish Nanayakkara*) [5], which was quite relevant to the given tasks and we were taught *Ja*va programming language in the first year at university, neither of the team members has ever worked with *NXT* kits or in *NXC* language. Therefore to solve those problems we had to not only research the algorithmic and hardware solutions, but also the *NXT* platform and *NXC* programming language.

2 Team Organisation and Team Management

Having some experience in working in a team of several people in the past I have realised that the success of failure of this group project will be very much influenced by team organisation as well as team management. And because I understood this importance I have decided to take the role of the team leader. During our first meeting I have suggested myself as a team leader and both my team mates agreed.

During the first meeting we have also decided that *Joshua Manuncia* is going to be the team secretary, however because he was absent from several following meetings, this role has been very early transferred to *Oluwasheun Adebari*. To ensure that minutes from each meeting meet certain standards I have written the minutes from the first meeting to show the team secretary what should minutes from each meeting contain. From the second meeting on I have not been involved in writing the minutes. However as the team leader I have proof read and approved every minutes before they were submitted, to ensure the accuracy and quality of each recorded minutes.

For both of the projects I have led the team to work following a simple iterative weekly development strategy, very similar to *Agile Software Development* [6]. The idea of this development strategy is that the project moves very fast from a new idea through a research phase to the development and testing. This means that for both the hardware and software parts of our tasks we have started with some idea and build on that with small iterative improvements to make our solutions better.

Because for both tasks I have had some sort of idea how to approach the problems I have for each week always selected one or more topics to be researched. Outside the meetings during the week each team member would do individual research and then we would bring the results together during the next meeting. If there was just one topic, it was researched by every team member, if there were more topics, those were split among ourselves.

A typical weekly meeting consisted of firstly checking that our current solution still works, then discussion about the research the team members have done and suggesting and experimenting with the ideas and results which came out of our research. In some cases (mainly the hardware design for the ball collecting robot) the result was a product of experimenting rather than the research, but overall the majority of the work was based on research.

6th Feb	Hardware Design Ready
13th Feb	Testable Ball Collecting Mechanism
20th Feb	Final Version of Ball Collecting Mechanism Working on Unloading Mechanism
27th Feb	Working on Search and Turning Algorithm
6th Mar	Search and Turning Algorithm Completed
13th Mar	Final Version of Hardware and Software Done
20th Mar	Presentation and Robot Ready for Competition

Table 1. Schedule for second term.

Since the first term problem has proven to be easier to solve, the team management for this problem was also easier. Because it was obvious that the second term task was going to be much more challenging, I have during the first meeting of the second term together with my team members created a time schedule based on which we have organised our meetings (Table 1). We have set dates when certain parts of the project were supposed to be finished and if we were on time, we met just once that week. However if we saw that we were not on time, we would meet twice or three times a week, just to make sure we finish that particular part of the project before our set deadline.

It was up to me as the team leader to organise all team meetings and therefore it was also me who made sure that we were on our schedule.

I feel that I have ensured that the project was managed accordingly and overall I think that I have contributed to the project quite a lot as the team leader.

3 Line Following Task

We have began working on this problem by creating a simple line following robot (using the provided instructions with the *NXT* kit), which featured the programmable brick, two wheels with motors and one passive caster wheel and one light sensor.

Our first software solution was a simple propositional controller, which used the light sensor to measure a light value and then move the robot accordingly. This was simple and reliable solution, but was not working properly for more complicated tracks (with perpendicular turns).

Because at that point we had been taught about the basic concepts of *PID* controllers at the *Adaptive and Robotics Systems* module [5], I have decided that we should research *PID*. I have given my team this task and we have for the several weeks tried to research and implement a proper *PID*. We have been able to find a very informative source [7], but unfortunately we were not able to find a proper *PID* calibration values for the controller and so we have decided to move on to another solutions.

Whilst trying to improve our initial version with proportional controller, I had an idea to use two light sensors on each side of the followed line, instead of just one sensor on the edge of the line. The software algorithm for this build was pretty easy - whenever one of the sensors hit a black line, the robot started turning until both sensors were again on white. When the sensor were placed further apart the robot could go relatively fast and be still able to manage to make quite sharp turns.

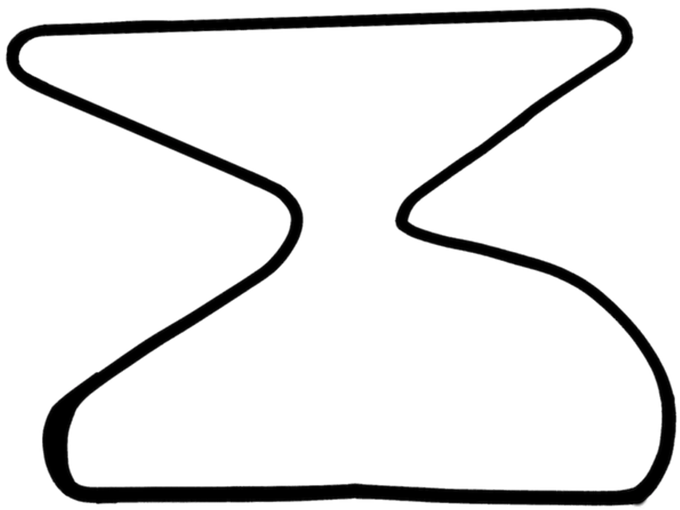

Figure 1. Long track with round corners.

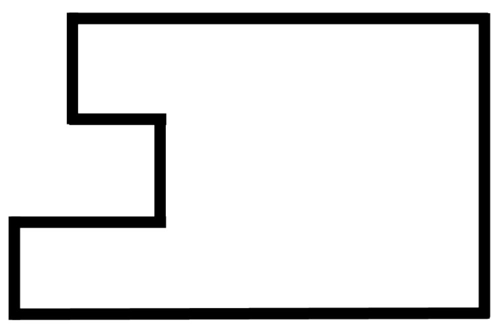

Figure 2. Short track with perpendicular corners.

Two weeks before the final race for the first task, we were given the two tracks, on which the race would happen, to test on. One long track with round turns (see FIGURE 1) and one short track with perpendicular turns (see FIGURE 2). Our two sensor solution performed very well on the long track, however was not accurate on the short track and therefore had to be made slower. I have approached our project supervisor *Dr Hongbin Liu* to ask which solution would be better to feature during our race — whether the fast but not always accurate two sensor solution, or the always accurate but slow solution. *Dr Liu* suggested that the ideal solution would be to combine both. With this idea I have came up with and implemented a simple timing algorithm, which run the two sensor solution for the part of the short track, where there were no sharp turns, and then switched to the proportional controller for the sharp turns. This switching algorithm also required us to equip the robot with three light sensors in total and then have the original two sensor algorithm utilise the two side sensors and the original proportional algorithm to utilise the middle sensor. This resulted in accurate and fast solution.

During the whole first term I have worked as the only programmer in our team. This was mainly because the programming part was not difficult to implement and therefore we did not need more than one programmer for this problem. However even though the code was written exclusively by me (as I was the one with the most programming experience), the ideas came from all members of our team and their research.

At the end of the term we have taken part in the race with our robot. Our robot performed well and won the first place, as it was faster than the other robot on both the short and the long tracks.

Figure 3. Line follower - final hardware design.

Overall I was very happy with the way how we worked on the first task and since it was not very difficult we did not need that much time to tackle this problem. The hardware design was not really challenging (as Figure 3 shows), as we have used the default build from the provided instructions with just a minor modifications, so the majority our the work was focused on the software part. I think that I have contributed fair amount of ideas towards the software program and also as a programmer I have implemented not only my ideas, but ideas from all members.

4 Ball Collection Task

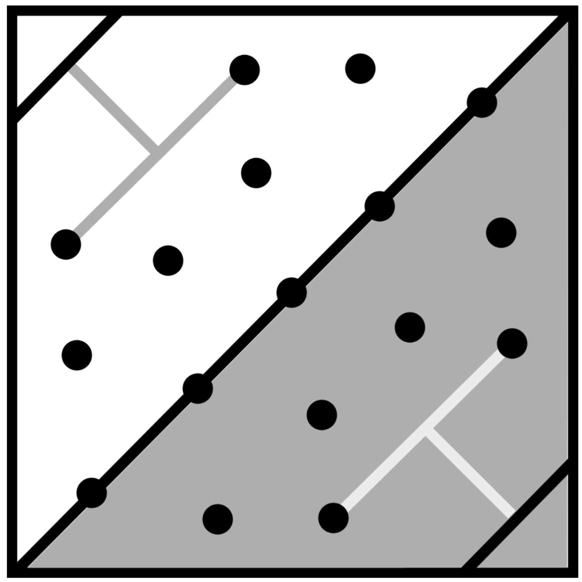

Figure 4.1. A diagram of a map for the ball collection.

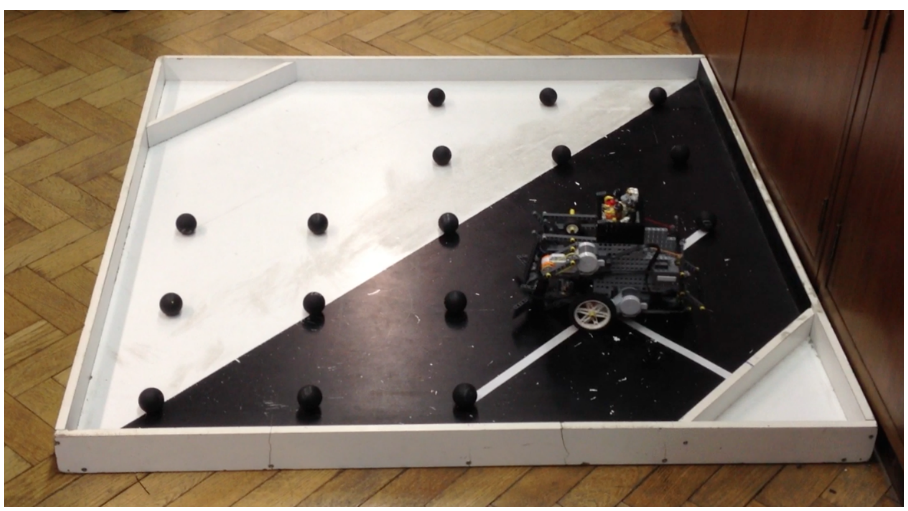

Figure 4.2. A photo of a map for the ball collection.

Before we began working on the second term task, we have been given instructions according to which we were supposed to create the robot. We were given a map (Figure 4) with balls at known positions. At the end of the term we were supposed to compete against the other team. Each team's robot would start in an opposite corner. Robots were supposed to collect and deposit more balls than the opponent. The robot which would manage to do that would be the winner.

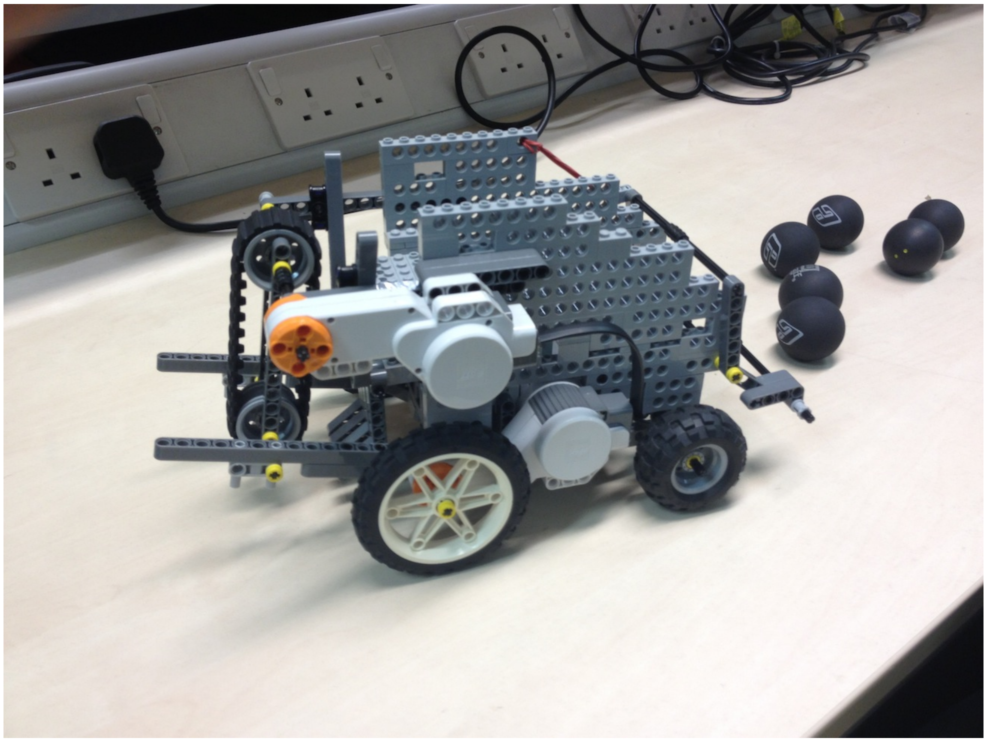

Figure 5. First ball collecting working prototype.

As mentioned in the team organisation and management section of this report, this problem seemed as more challenging than line following and therefore we have set a schedule (Table 1). As the team leader I have decided to start with the hardware design and build first, because I have realised that the software program will be essentially tailored to the hardware build. We have started by researching different existing documented ball collecting robots in Lego *NXT* on the Internet and experimenting at the same time. We have thought about building some sort of a ball grabber, but later we left this idea, as this mechanism would not be very reliable. When researching existing builds we have noticed that there are solutions using converter belts for bringing balls to a container. *Olu* has suggested using this approach and we have agreed on this. We tried several ways of doing this, but overall did not produce a functioning mechanism. My colleagues had to leave that meeting, and because I wanted to stay on schedule I have decided to stay late in the lab and work on the robot alone until a working prototype was created. I have managed to create a working prototype by the end of the day, with functioning ball collecting and ball unloading mechanisms (see Figure 5).

This prototype served as a strong foundation for our robot. Even though the first working prototype was a result of my individual work, the final hardware design is very much a result of team cooperation as we have made many improvements along the way, many of those suggested by *Olu* and many others by *Josh*. Overall I think that we have all contributed to the hardware design reasonable amount of work and collectively we have been able to create very sturdy and reliable hardware for our robot.

After we have created our hardware we have then moved on to the software. I have, as the team leader, identified two possible solutions to the given problem — either to create an adaptable algorithm, which would utilise the provided sensors, or a hardcoded preprogrammed route.

We have been given a set of sensors — *light sensor, touch sensor, ultrasonic sensor, accelerometer and gyroscope*. We have all made an initial research to find ways how to utilise those sensors in adaptive ball searching and after this initial research we have agreed that the only sensor which could be used for the ball searching is the ultrasonic sensor. At this point we have written down all the features our robot needed to have and we have split those amongst ourselves. Those features were *ball collecting, ball counting, ball unloading, turning specified angle, going straight forward and backwards and going certain distance.*

Each member has picked several of those features and then made research on those.

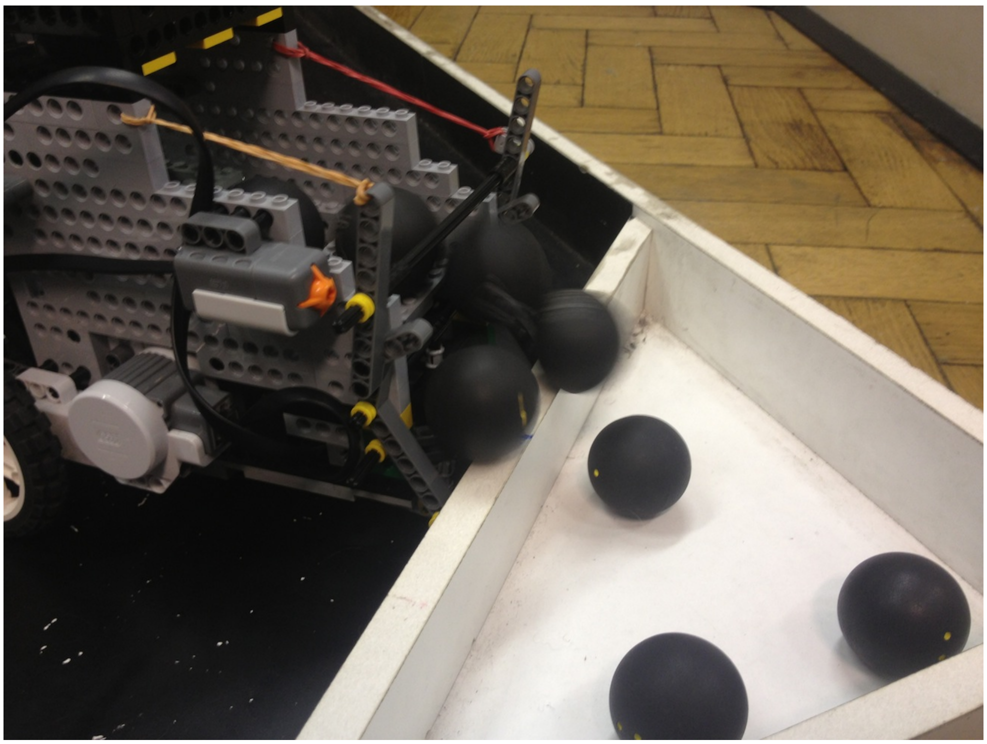

Figure 6. Ball unloading mechanism.

I have done research on ball counting and ball unloading. Initially I have decided to use light sensor for identifying collected balls, however because of unreliability of the light sensor (because of shadows, lights being on / off etc.) I have decided to use a touch sensor instead. This resulted in much more reliable way of counting balls. The ball unloading algorithm plays very much supporting role to the hardware design of the robot and it's unloading mechanism. The idea is that the robot uses kinetic energy to push itself against a wall of the container, where it needs to deposit the balls, and this causes the ball barrier to move and release all balls to the container (see Figure 6).

Olu has worked on motor synchronisation (to make sure that the robot can go in perfect straight line) and turning. *Josh* has worked on ball searching and ability for the robot to go accurate distance (specified in cm). Unfortunately after the Josh's research and our experiments with the ultrasonic sensor, we have concluded that the readings from the sensor are not accurate and reliable enough and therefore we have decided to implement the preprogrammed route instead.

This meant that we needed to focus more on reliability of our control functions, meaning that we needed to be sure that when we told our robot to turn 90° left, it needed to turn that angle, if it did not, the preprogrammed route would not be followed and therefore our solution would not work. We have spent many hours calibrating our control functions to make sure the robot does follow the given instructions as closely as possible.

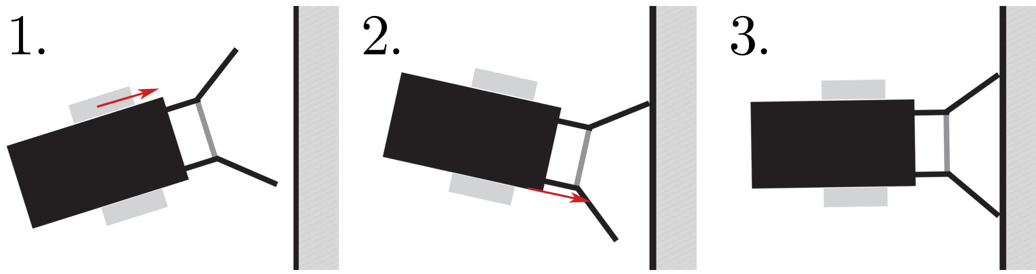

Figure 7. Wall aligning algorithm.

However because of friction, imperfections of the *NXT* Lego kit and different battery levels at different times, we have always encountered some degree of deviations from the given instructions. We were therefore looking for some fixing mechanism, which would bring our robot back to the original route, if any deviation occurred. I have come up with an idea of using walls of the map to align the robot. The idea follows a simple mechanism, whenever the robot is close to a wall, a torque is applied first to one wheel for a split second and then to the second wheel. This causes the robot to get "pushed" against the wall and therefore to align perpendicular with it, even if originally the robot was under a different angle (see Figure 7). I have implemented this algorithm and it has been featured several times in our solution.

Throughout the second term I was still working as a lead programmer for our team, so however not all code was written by my, I have helped to implement most of it and I have made sure that code from all members gets integrated into one final working solution.

Because we have set the schedule (Table 1) for the second term and because I have been able to organise our team to keep up with the schedule, we have finished with the majority of works two weeks before the final competition and we were able to spend last two weeks tweaking and calibrating our program as well as preparing for the final presentation. I have as the leader made sure that all members from our team had chance to practise the presentation, so that we would all be ready to describe how we made our robots.

At the end of the term we have taken part at the final competition and I have to report that we have again won, collecting and depositing the majority of 9 balls (out of total of 17), our opponents collected and deposited only 2 balls. Also I feel that we have addressed all topics necessary in our presentation and that we have been able to explain the whole process of creation of our ball collection robot.

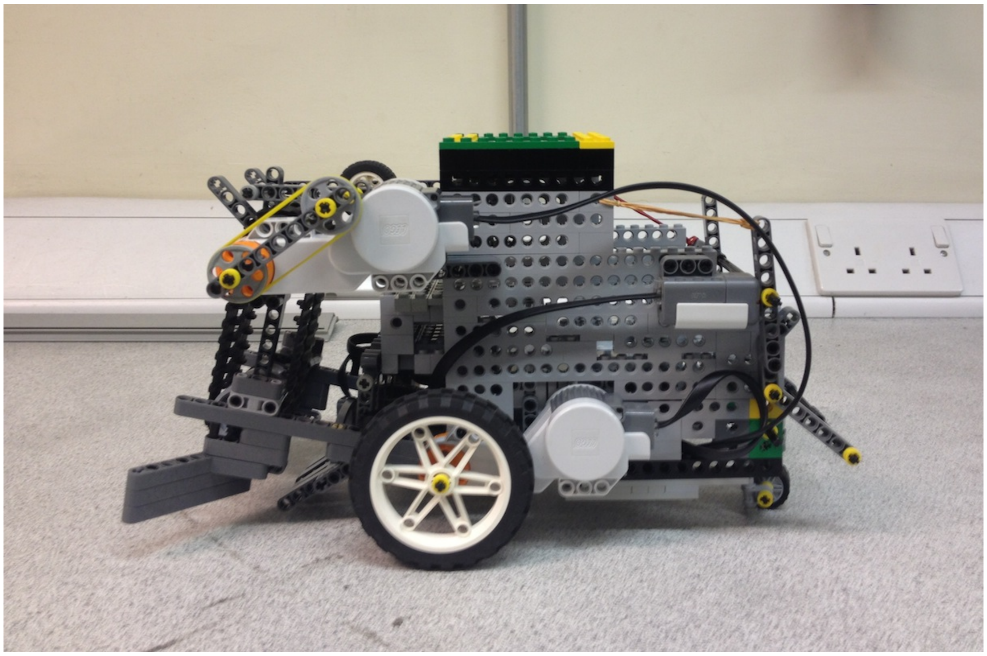

Figure 8.1. Final hardware design for ball collecting.

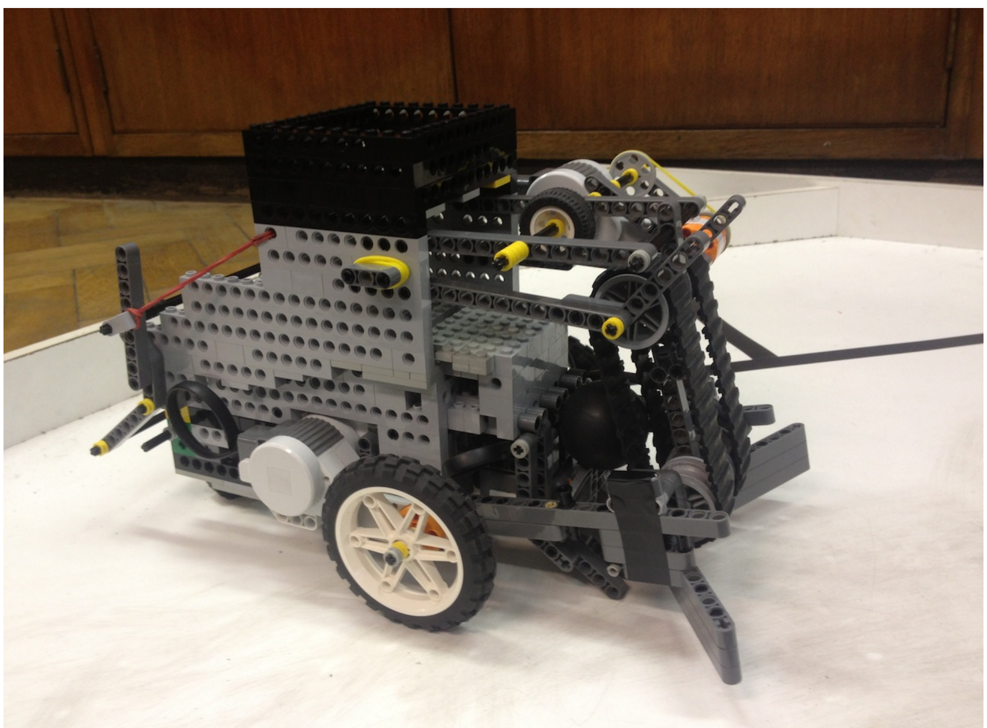

Figure 8.2. Final robot collecting a ball.

Overall I think that our presented robot's hardware (see FIGURE 8) and software were both products of our team work and we have been able to build on each others work, improve it and bring all our research as well as ideas together to very well functioning hardware and reliable software.

5 Discussion

5.1 Limitations

Line follower

Our two strategies which have been developed are quite reliable. However the two sensor algorithm has it's limitations on a track with sharper turns and the proportional controller is accurate and reliable, but tends to be slow.

Even though we have tried to implement a *PID* controller, we have not been successful. This was not a major issue, but I think that if implemented and set properly *PID* would provide much more efficient and faster way how to follow a line.

Ball collector

I think that the hardware build we have designed is very sturdy, reliable and very much a good solution to the given task of ball collection.

However the chosen algorithmic solution is working only for the given map with balls at known positions. This means that the robot is not suitable for use in unknown environments.

5.2 Future Work

Line follower

As mentioned in the previous section, the obvious improvement for our line follower would be proper implementation and setting of a *PID* controller.

Ball collector

If our solution would be further developed, the first step would be to fit the hardware with more sensors, which we did not have available (*compass, more reliable sonar sensor*, etc.), then those sensors could be utilised to implement a mapping algorithm with ball searching. The robot would then map a room in which it would be placed in, find all balls and then collect and deposit them in the designated area.

6 Conclusion

6.1 Results

We have done quite a lot of research for both given tasks, which has allowed to find the appropriate solutions for our problems.

Despite having to solve some challenging and difficult problems, we have created two functioning robots for both tasks, which have met all the requirements and have been able to perform with low error occurrence and also outperform our opponents' robots in both contests.

I believe that we have therefore been very successful in both given tasks.

6.2 Team work

Over the time we have managed to develop very good team coordination and we're able to work very well together.

We have been able to keep up with the set schedule as well as record meeting minutes for every single week.

6.3 Individual contribution

I was helping my peers as the leader team. I organised our weekly meetings as well as made sure that we are on schedule. I would like to think that I was able to lead my team as a good team leader and that they were happy with me as well as I was happy with them.

I have also been part of the team as the chief programmer. In this role I have taken care of the majority of programming and I helped both of my team mates to implement their ideas and results of their research.

I have designed and built the first hardware prototype (FIGURE 5) of the ball collecting robot and furthermore I have worked together with my team mates on all hardware improvements made on the initial design.

As well as all other members of my team I have also done research on parts of the problems and I have helped to bring the results and solutions together to the final product.

References

[1] Lego. Mindstorms kit. *http://mindstorms.lego.com/en-gb/default.aspx*, accessed on 29/03/2013.

[2] Wikipedia. Lego Mindstorms NXT. *http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT*, accessed on 29/03/2013.

[3] SourceForge. Bricx Command Center. *http://bricxcc.sourceforge.net*, accessed on 29/03/2013.

[4] Wikipedia. Not eXactly C. *http://en.wikipedia.org/wiki/Not_eXactly_C*, accessed on 29/03/2013.

[5] King's College London. Adaptive and Robotics Systems. *http://kcl.ac.uk/nms/depts/informatics/study/current/handbook/Progs/Modules/5CCS2ARS.aspx*, accessed on 29/03/2013.

[6] Wikipedia. Agile software development. *http://en.wikipedia.org/wiki/Agile_software_development*, accessed on 29/03/2013.

[7] J. Sluka. A PID Controller For Lego Mindstorms Robots. *http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html*, accessed on 29/03/2013.