HCL Embedded real-time system with applied machine learning

Embedded systems

Aalborg University

Group SW501E18

Title:

Software 5: "HCL Embedded real-time system with applied machine learning"

Group SW501E18

4. September - 2018

AALBORG UNIVERSITY
STUDENT REPORT

Semester:
5. semester
Project theme:
Embedded Systems
Project duration:
04/09/2018 - 20/12/2018
ECTS:
15
Supervisor:
Thomas Bøgholm

Authors:

Casper Weiss Bang

Daniel Moesgaard Andersen

Frederik Spang Thomsen

Nichlas Ørts Lisby

Thomas Højriis Knudsen

Thomas Lundsgaard Hansen

Abstract:
In the Danish educational system, the Arduino platform is often used to introduce students to computer science. The Arduino's hardware is relatively simple and easy to grasp. However, the Arduino language can often be difficult for novices. While the Arduino language is a simplified subset of `C++`, there are still parts of the language that are difficult to comprehend for novices.
This report elaborates and explains the development of the HCL language, a programming language designed to ease the introduction to programming for students and novices, whilst also implementing high-order functionality on the Arduino. HCL does this by emphasizing a resemblance to the English language. A compiler for HCL was developed, as part of the project. The compiler is written in Kotlin, by hand, and compiles to `C++` code.

Number of pages: 17.

Preface

This report is written during the fifth semester of the bachelor's degree in Software at Aalborg University. It is written for the study board of computer science at the School of Information and Communication Technology. The report is written with the guidance of one supervisor.

The topic of the semester is 'Embedded Systems'.

The system to be developed was chosen by the project group.

Terms and abbreviations used in the report:

TM : Turing Machine
NTH : Nice to Have
NXT : Lego Mindstorms NXT 2.0
EV3 : Lego Mindstorms EV3

Contents

1 Introduction

Tracking airborne objects can be quite difficult. This difficulty has inspired different sports and hobbies, including skeet shooting. Hitting a target requires the tracking of it, it's velocity and direction, as well as the calculation or prediction of its future trajectory. The trajectory can be impacted by a number of different features, including gravity, wind, initial force, any artificial acceleration. Furthermore, if the object is intelligent in anyway, the path might change procedurally.

Tracking of airborne objects has multiple different applications.

1.0.1 Airports

Estimated thirteen thousand "bird strikes" happen each year, which is the event of a bird colliding with a vehicle, where the majority is aircrafts[1][2]. Even though 65% are estimated to cause little damage, the damages are estimated to cost 1.2 billion dollars each year, world-wide [3]. The majority of these accidents happen at low altitude, during landing or takeoff[3]. Airports utilize different methods of bird control, including killing, trapping, poison and lights[4]. Shooting birds is one of the most common methods, and has been observed to be quite effective, while also reinforcing other methods. This both kills a singular bird, but also scares away other birds. However, it can be expensive in both salaries and effort.

1.0.2 Military

In a military setting hitting a moving airborne target is relevant in a defence context, for instance destroying incoming projectiles or shooting down hostile planes. Compared to the airport-case, the problem can arise at a variety of locations, both on vehicles, boats, planes and tanks, but also on stationary military installations. The American military developed the Phalanx CIWS, Close-in weapon system, which is a radar-guided 20mm Gatling-style rotary cannon, mounted on a base that moves on two axes[5]. These are mounted on ships and can shoot incoming missiles and other airborne objects deemed a threat. The Dutch also developed a similar system called the Goalkeeper CIWS. These are both autonomous.

1.0.3 Initial Problem

Tracking an airborne object has multiple different problem areas that makes it an interesting domain. This makes it interesting to research, and this report will try to look into ways of solving this problem on a smaller scale, in clinical examples, to research the different parameters. Before hitting an airborne target, it is important to estimate its trajectory, and the problem can therefore be simplified to two subproblems.

1. Calculating how many seconds it takes before a projectile can reach the target.
2. Calculating where the target will be in T seconds.

Combined, these two problems means finding the spot where the travel-time of the target meets the travel-time of the projectile.

This report is build on courses in both real-time systems as well as machine learning, and this problem works in the context of both. In regards to machine intelligence, the problem of tracking an airborne target has potential for intelligent analysis by an artificial agent. Likewise, it is a time-critical task to move a weapon so that it aims at the target and hits the target, before the target is out of range or hits the ground.

As hitting the target with a projectile introduces a lot of unknown variables, this report will focus on tracking the target with a laser, as the travel time is near instant (based on the speed of light), and it's less affected by gravity. The report will use more simplistic targets, to remove unpredictable parameters, and to remove the possibility of harming any living creatures.

These considerations creates the following initial problem, whereas the analysis will create a more in-depth set of requirements and a concrete problem statement.

How can a software system track a moving airborne projectile

Analysis 2

2.1 Embedded platform

The goal of the 5th semester of the bachelor in Software, is to develop an embedded system. Therefore it is relevant to look into the different relevant platforms. The university supplies the student groups with LEGO Mindstorm, which would make it an ideal option, however it is is still interesting to look into alternatives.

2.1.1 LEGO Mindstorms

LEGO Mindstorms is a programmable computer kit, made by the LEGO Group. The platform is based around a central control computer, called the Intelligent Brick. It also includes a variety of sensors, motors and connection cables.

The LEGO Mindstorm platform, comes with a graphical drag and drop interface. However, various different compilers for many different languages exist. As such the choice between different generations, does not come down to language preferences.

Each group is supplied with a LEGO kit, consisting of both a LEGO Mindstorms NXT 2.0 kit and a LEGO Mindstorms EV3 kit, as well as an assortment of sensors, motors and LEGOs. On top of this, it is also possible to order other platforms and sensors.

LEGO Mindstorms NXT 2.0 The LEGO Mindstorms NXT 2.0 (NXT) is the second generation LEGO Mindstorms kit. The NXT Intelligent Brick is a micro computer based on the 32-bit 48 MHz ARM7 microprocessor, with 256 kilobytes of flash memory and 64 kilobytes of RAM [6] [7]. It includes a variety of sensors, such as touch, sound, light, and distance sensors, as well as servo motors and lamps.

A potential major disadvantage of the NXT 2.0 is it age. The platform was released in 2006, meaning that the software might cause problems regarding implementation in modern day software solutions.

LEGO Mindstorms EV3 The LEGO Mindstorms EV3 (EV3) is the third generation of the LEGO Mindstorms line. The EV3 mainly improves on the specifications of the second generation, adding a significantly more powerful 300MHz ARM9 processor, running Linux, 16 megabytes of flash memory, which is extendable with a microSDHC card and 64 megabytes of RAM [8]. Furthermore, it adds remote control, WiFi capabilities, and more recently updated compilers.

It is also fully backwards-compatible with the NXT sensors [9].

Both of are a bit more akin to general purpose computers, designed for small electronics projects and both platforms include different models, with some being more powerful than others or having different features that further set them apart.

Arduino

The Arduino is a single-board computer platform, designed for small electronics projects by being easy to use from a hardware and software perspective. The Arduino Corporation developed a language, a subset of `C++`, which is most often used when working with the platform. The primary advantage of the platform is that it is easy to work with and that there are so many different boards, all of which differ in performance, meaning that one that matches our particular requirements, most likely exists.

The IO possibilities of the Arduino are GPIO[1] pins, which can connect to most types of sensors and motors, as these are, as implied by the name, general purpose.

The Arduino-ecosystem is comprised of different boards, all of which are built for different purposes. Differences being in processing power, as well as pins. The Arduino Uno is most commonly used board in the ecosystem [10], and has a 12 MHz ATmega328P processor and 2 kilobytes of RAM. A larger version, the Arduino Due has an 84 MHz processor and 96 kilobytes of RAM [11], and more input/output pins and is meant for larger scale Arduino projects.

Raspberry Pi

The Raspberry Pi is a single-board computer eco-system which, like the Arduino, consists of small computers with GPIO pins. It differs from the Arduino, by having a full operating system, and generally being more powerful. Additionally, the majority of the Raspberry Pi computers have more processing power and memory, and the newer versions include a Wi-Fi module. They also come with a larger variety of ports - both USB, HDMI and jack, adding more possibilities for connectivity.

The newest version, Raspberry Pi 3 B+, features a 1.4 GHz quad-core CPU, with 1 gigabyte of RAM. They released a light-weight version, called the Raspberry Pi Zero, which trades performance, having only a 1 GHz single-core CPU and 512 megabytes of RAM, for a smaller footprint, being less power consuming, and a lower cost.

[1] General Purpose Input Output

2.1.2 Takeaways

There are multiple platforms on which to build an embedded system. This data is summed up in the following table.

The Raspberry Pi system can be quite powerful a project regarding embedded systems, which can also be said about the EV3. This can be a disadvantage, as we aren't fully in control of the execution order.

The Arduino and NXT both have less processing power than the Raspberry Pi and EV3, and thus making them more interesting choices, by limiting the hardware capabilities. The disadvantage of the Arduino has fewer supported programming languages, and the fact that the GPIO ports make for hardware-related challenges, which is not the purpose of this semester. Unlike the Arduino, the NXT offers easy to use hardware ports, and large variety of sensors and motors. As the NXT is supplied by the university it makes for the obvious choice, especially considering the points provided in this section.

2.2 Sensors and motors for Lego NXT

The Lego NXT series offers a large variety of sensors and motors available for use, ranging from light and sound sensors to dual range, triple zone infrared obstacle detectors. Due to the amount of available components, it is pertinent to further investigate the advantages and disadvantages of each component to find the best applicable ones for the project.
In this section, the focus will be held on the basic functionality of the available components.

2.2.1 Ultrasonic Sensor

The primary purpose of an ultrasonic sensor is to determine distance from the origin to one or more objects using ultrasonic waves. To do this, the sensor head sends out an ultrasonic wave and receives the wave reflected back after hitting the surface of the target object. Using the time interval elapsed between sending and receiving the wave, the distance can be calculated.
The NXT Ultrasonic Sensor has an effective range of 1 to 250 centimeters with an accuracy of +/- 1 centimeter.

	Processor	MHz	RAM	Fl
LEGO MindStorms NXT 2.0	32-bit Atmel AT91SAM7S256 ARM7	48 MHz	64 KB	25
Lego MindStorms EV3	32-bit TI Sitara AM1808 ARM9	300 MHz	64 MB	16
Arduino Uno	8-bit AVR ATmega328	20 MHz	2 kB	32
Arduino Due	32-bit Atmel SAM3X8E	84 MHz	96 KB	51
Raspberry Pi 3 B+	64/32 bit ARM Cortex-A53 quad core	1.4 GHz	1 GB	M
Raspberry Pi Zero	32bit ARM1176ZF single core	1 GHz	512 MB	M

2.2.2 Touch sensor

As implied in the name, this type of sensor allows detection of touch. This allows for a variety of functionality such as enabling the machine detect if it collides with other objects.

2.2.3 Light sensor

The light sensor allows measuring the light levels by using an LED and a photo resistor. The sensor is capable of both measuring ambient light, such as the light levels of a room, and reflected light by using the built-in LED.

2.2.4 Dual range, triple zone infrared obstacle detector

The dual range, triple zone infrared obstacle detector, also known as NXTSumoEyes, is an obstacle sensor with an effective range of up to 20 centimeters using infrared beams. The major usability difference between the NXTSumoEyes and the ultrasonic sensor is that the NXTSumoEyes works with three individual detection zones of 20 degrees each, allowing for a more detailed response on the location of the detected object.

2.2.5 IRRD-T sensor

The infrared relative distance sensor, IRRD, measures the distance to an object with an effective range of up to 60 centimeters by measuring reflected infrared light, in a similar faction to the ultrasonic sensor.

2.2.6 HPMR Infrared Distance Sensor

The high precision medium range infrared distance sensor work in the same way as the IRRD-T sensor previously described.
However, this sensor allows detection of objects with a range of 10 to 80 centimeters, by using a Sharp GP20A02YK sensor, which is slightly more powerful than the sensor used in the IRRD-T sensor.

2.2.7 Motors

The NXT series offers a variety of motors in two categories: DC motors and servo motors. The DC motors are regular powered motors that can simply be turned on and off, while the servo motors have the functionality of controlling the rotation by a given degree.
The servo motors have a speed of 160-170 RPM, and a rotation precision of 1 degree. In comparison, the DC motor has a maximum speed of 380 RPM, making it faster but allowing less control of the rotation.

2.2.8 Angle sensor

The angle sensor measures axle rotation position and rotation speed. It allows measuring both the absolute angle, meaning the rotation of an axle from 0 to 359 degrees with an accuracy of 1 degree.
Likewise, it can measure an accumulated angle, which is the accumulated multiple rotation angle measured since it was last reset. Finally the angle sensor can calculate the current RPM with a rotation rate from 1 to 1000 RPM, with an accuracy of 1 RPM.

2.3 Managing systems in Real Time

On of the examples that this project was build upon, was defense systems on boats, where the device has to shoot down missiles. In this case it is rather important that the missile is shot down before it reaches its target. The other example, was shooting birds in airports. Both of these examples are time critical in the sense that the given device has to continuously follow a moving target, and the target obviously will not move slower because of latency in the CPU.

So even in the case of hitting a ball, a rather mundane target relatively, it is important to build an efficient system, especially when using a small architecture, such as the NXT. This is done by using real time system principles. A couple of different principles and specific problems, should be considered regarding this issue.

Deadlines As the device has to predict where to aim the laser, it is important to know exactly how long it takes for the device to react, within a reasonable degree of scientific certainty, so the laser is going to lead the target, rather than follow it. The deadline, in this case, is the target being out of range, and the time frame is a function based on the speed and acceleration of the target. Deadlines can also be more critical in as data might lose relevance and become obsolete. Depending on how the object moves, and how predictable the target is, it is important to make the react-time as quick as possible, as a larger time frame might make the predicted location less certain.

Schedulability It is quite important to be able to schedule the relevant events; processing the image, predicting the trajectory, and moving the motors, before the target is out of sight. To do this certain analytic decisions need to be made, regarding the time frame of these actions. It is also quite relevant regarding the order in which to do the different tasks, and estimate what has the highest relevance.

Takeaway Looking into how to optimize the different processes is important. Without considering the order in which to do tasks, and the expected time frame of a task, it is impossible to interact with a moving targets and a changing world.

Therefore, the principles of Real Time Systems, are quite relevant to the problems of this project.

Theory 3

4 Implementation

Bibliography

[1] Wikipedia; Various authors, "Bird strike," https://en.wikipedia.org/wiki/Bird_strike, 2018, [Online; Accessed: 18-09-2018].

[2] US Department of Transportation, "Wildlife strike report 1990-2015," https://wildlife.faa.gov/downloads/Wildlife-Strike-Report-1990-2015.pdf, 2015, [Online; Accessed: 18-09-2018].

[3] John R. Allan, Alex P. Orosz, "The costs of birdstrikes to commercial aviation," http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1001&context=birdstrike2001, 2011, [Online; Accessed: 18-09-2018].

[4] Programmadirectie Ontwikkeling Nationale Luchthaven, "Bird control at airports," http://www2.vlieghinder.nl/naslagdocs/CDrom/REGELS_SCHIPHOL/2.3_TNL/4.3.4.1_Bird_control_at_airports.pdf, 1999, [Online; Accessed: 18-09-2018].

[5] Wikipedia; Various authors, "Phalanx ciws," https://en.wikipedia.org/wiki/Phalanx_CIWS, 1999, [Online; Accessed: 18-09-2018].

[6] Lego, "Nxt user guide," https://www.generationrobots.com/media/Lego-Mindstorms-NXT-Education-Kit.pdf, 2018, [Online; Accessed: 2018-16-09].

[7] Botbench, "Comparing the nxt and ev3 bricks," https://www.generationrobots.com/media/Lego-Mindstorms-NXT-Education-Kit.pdf, 2018, [Online; Accessed: 2018-24-09].

[8] Lego, "Ev3 user guide," https://le-www-live-s.legocdn.com/ev3/userguide/1.4.0/ev3_userguide_enus.pdf, 2018, [Online; Accessed: 2018-16-09].

[9] Legoengineering.com, "Ev3 software compatibility with nxt," http://www.legoengineering.com/ev3-software-compatibility-with-nxt/, 2018, [Online; Accessed: 2018-24-09].

[10] Arduino, "Arduino uno rev3," https://store.arduino.cc/arduino-uno-rev3, 2018, [Online; Accessed: 2018-24-09].

[11]