Integrating real-time image processing into rapid prototype robotics

[Project Short Paper] *

Jean-Clive David Bailey
Rhodes University
P.O. Box 94
Grahamstown, 6140
g11b3549@campus.ru.ac.za

ABSTRACT

The aim of this project was to successfully create a modular robotics system using the rapid prototyping design philosophy and incorporate an external field of computer science knowledge into the system. The driving focus is to evaluate how an independently built robot can be used to solve a real world problem or perform a specific task without the constraint of specialized programming environments that limits experimental development. The goal is to observe how these methodologies impact the performance of the robot, the difficulty of producing goal-driven robotics outside a controlled teaching environment, and the complexity of integrating a useful subsystem.

CCS Concepts

•**Computer systems organization** → **Robotic control;** •**Computing methodologies** → *Image processing;* •**Software and its engineering** → Software prototyping;

Keywords

Robotics, Image Processing, Design, Prototyping

1. INTRODUCTION

The availability of low cost and easy-to-use products enables any teenager or adult with no formal engineering education to be able to create simple, quickly assembled robots to perform rudimentary tasks [6]. Tools like LEGO Mindstorm and the Python-based Pyro are suitable as research-level robotics hardware and methodologies that are accessible to anyone within a computer science faculty who may not have robotics experience [3]. With this as a baseline, this paper explores integrating a rapid prototype development cycle in conjunction with real-time image processing. The combination of the two fields allows for flexible designs to solve problems of varying degrees of difficulty. In February 2014, an Atmel based small-scale robotics project using an Arduino Uno with ATmega based micro controllers was designed with the simple task of writing the current time using a whiteboard marker on a surface and then with the same mechanic erase it [2]. With only three mechanical servos and a Computer Aided Design (CAD) 3-Dimensional (3D) printed body, this small scale project would be an ideal way to test the true nature of rapid robotic prototyping. The Plot Clock will serve as a comparable system for this paper. The goal of this research is to replicate the design, report the ease of rapid prototyping and the difficulties of integrating the final design with real-time image processing.

*A full version of this paper is available at http://www.cs.ru.ac.za/research/g11B3549/

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. RELATED WORK

2.1 Existing Plot Clock robot

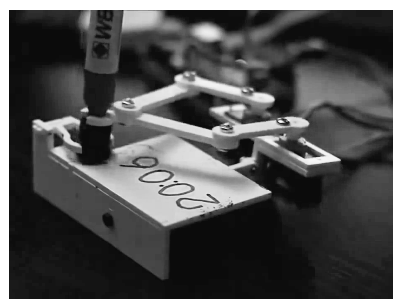

Figure 1: Heberlein's Plot Clock

The Plot Clock is an open-source hobby kit designed by Johannes Heberlein and published online[1]. The entire kit, which includes .ino source code and Computer Aided Design (CAD) templates, is free to download and can be built using an Arduino Uno, three servos, and a whiteboard marker[7]. The Arduino is programmed in C. The purpose of the clock is to have a mechanical drawing arm pick up the board marker and use it to plot the current time on an drawing surface and erase it with a wiping cloth. This robotic system makes use of a loop to maneuver the arm to each digit position in the clock and plot it accordingly.

[1]Thingiverse – http://www.thingiverse.com/

2.2 LEGO Mindstorms NXT

The LEGO Mindstorms systems, a joint venture between the LEGO®Corporation and the Massachusetts Institute of Technology (MIT), is a series of programmable robotic kits predominantly utilized as a teaching tool in introductory programming courses. The second generation released in June 2006 and is called the NXT series, which replaced the first series called Robotic Command eXplorers (RCX) [10]. The Mindstorms Kit is a popular choice among robotics enthusiasts and academia since it provides desirable properties as opposed to competing units like Adept MobileRobots's Pioneer robots [1] and MIT's HandyBoard and Super Cricket [14].Klassner and Anderson [9] demonstrated that LEGO Mindstorms became a suitable platform for college students to investigate a broad range of topics and they had specifically chosen the LEGO Mindstorms for a few distinct reasons – three are applicable to this project:

1. **Cost:** A single Mindstorms kit, with 750 construction pieces, sensors, and programmable hardware, costed approximately $200. At the time this was a quarter of the HandyBoard-based robot kit and a mere tenth of the MobileRobot kit.

2. **Flexibility:** The Mindstorms platform supports a suite of reusable snap-together sensors, effectors, building blocks and control unit that can serve as a basis for a variety of projects.

3. **Professional Curiosity:** Other liberal-arts college instructors cited that the Mindstorms kit was a great source for robot chassis construction.

2.3 Raspberry Pi Robotics

The Raspberry Pi (RPi) is a low cost, credit-card sized computer with multiple peripheral ports that allow connection to traditional Human Interaction Components like a mouse, a keyboard and a monitor, but it additionally hosts a General Purpose Input Output connector that interfaces with electronic circuits [15] – the quintessential piece for building a Raspberry Pi based robot. Over the last five years the Raspberry Pi has been released as four versions, each building upon the strengths of the previous generation and providing a multitude of features. Several research papers incorporate the use of RPi units, and have been found to be suitable micro computers in both analytic and control mechanisms [12, 4, 5].

3. THE PROBLEM

The aim of this paper is to outline the process of successfully creating a modular robotics system using the rapid prototyping design philosophy and incorporate an external field of computer science knowledge into the system. The driving focus is to evaluate how an independently built robot can be used to solve a real world problem or perform a specific task without the constraint of specialized programming environments that limits experimental development. The goal is to observe how these methodologies impact the performance of the robot, the difficulty of producing goal-driven robotics outside a controlled teaching environment, and the complexity of integrating a useful subsystem. In this paper the following research questions are raised:

1. Will a replica robot constructed from alternative materials and environment have the same qualitative output as the original?

2. Can a bespoke robotics system be altered to suit secondary purposes, such as real-time image processing?

3. How difficult is it to implement a rapidly developed subsystem into an working robotic system?

4. What design choices will be made during the implementation to ensure the streamlining of modularization?

3.1 The Solution

To allow enough time to develop and experiment with rapid prototyping, with the volatile nature of software development in mind, this project will be replicating an existing robot that was published on the Internet as an open source hobby kit in 2014. The original robot known as Plot Clock, will serve as baseline for the prototype and will act as an initial milestone for the project. The Plot Clock is a static program system, meaning it only follows predetermined instructions that it is hard wired to - this is where this project will improve the system by adding in an external field of knowledge: real-time image processing. By utilizing a web-cam and simple artificial intelligence (AI) this paper will expand and improve the original design using the rapid prototype and goal-driven design paradigms. This improvement may not be relatively new in terms of robotic AI, but could provide an interesting look into how such a system can be adapted to provoke solution refinement.

4. SOLUTION DETAILS

To facilitate the outlined approach, the Plot Clock will utilize a cyclic design paradigm to refine the initial working model into a fully functional unit. Each iteration seeks to improve the overall performance of the entire system, but predominantly the physical robot (1) and driving software (2) will be focused on, since these two layers encompass the majority of the entire project. The Plot Clock will need to perform the main function of writing the time on a drawing surface using two mechanical arms to manipulate a whiteboard marker.The physical design of the robot must attempt to emulate Heberlein's original concept as best as possible. Key considerations of how the robot functions will drive the shape. One particular choice will be how the components function differently to that of the original robot chassis, which is constructed entirely out of 3-Dimensional (3D) Printed plastic. It inculcates the question if the LEGO NXT bricks are able to match the style of Heberlein's Plot Clock.

The ability to differentiate between virtual measurements and actual measurements is another important reasoning behind a fair test into *how* the system performs. Evaluating the way in which the robot manipulates the whiteboard marker is an essential measurement to in determining the success of the system. One approach is to compare the written output to the actual digital representation source. For example, if the Plot Clock writes the digit '3', a test will compare the written result on paper compared to a digital copy of the case. There is an expectation of high inaccuracy at first. However, with the nature of rapid prototyping in mind, an iterative refinement cycle will narrow the gap in technology difference between this project and Heberlein's Plot Clock.

The Plot Clock was originally designed to only write the time inside a bespoke system loop – with the capability of moving the marker in three planar dimensions, there is an intent to broaden the original system scope. A key goal of this system is to allow users and future implementations to use the robot's design and software as a baseline for more experimentation. Therefore, the robot must be modular in nature, allowing a stream of input which it can execute without causing software or hardware faults.

Heberlein's original robot design will be expanded upon to integrate an image processing part as an interactive component. The system must be able to translate a feed of data in real-time and interact with the robotic system and have a distinct influence in output. This would be classified as successful if the robot responds to the data stream within one second of receiving data.

4.1 Plotting with mechanical arms

Creating an independent calligraphic robot using two drawing arms requires a two-step operation – each step acting as a gateway between two distinct planar transformations. From a given desired coordinate, the first step requires mapping both the X and Y coordinates to a scaled drawing dimension with exact coordinate points. The second step is transforming the scaled drawing points into two real angles that can be applied as rotational angles to the motors. The significance of each step's transitions is that it illustrates how the initial desired point is converted to an acceptable virtual point on a virtual geometric plane and then converted again into two real angles in the physical geometric plane. At this point, the real angles can be directly applied to the respective motors controlling the drawing arms, creating the desired arm shape in the physical world. In short, desired point → virtual point → two real angles → physical movement.

Conceptually it is impossible to draw any point in an unknown space without knowing the constraints of the geometrical plane, or surface. A true-negative, for example, would be applying the point (100,100) in a surface that ranges from X:0 → 50 and Y:0 → 200, since the X coordinate lies outside the available range. Similarly, another true negative would be applying a negative coordinate point in a pure positive surface. This signifies the importance of this step as its goal is to prevent incorrect input. Detecting such points is a trivial matter of evaluating each coordinate in relation to the upper and lower boundaries of the surface. A better solution would be to identify the boundaries of a set of known or unknown points and then apply a matrix transformation to every individual point. This way the entire set can be applied to the surface as virtual points.

4.1.1 *Mathematical formulae*

Several formulae were derived to simplify the logic behind the running code. These formulae progress a desired point to an actual true rotation angle. The known values are the lengths (mm) of each segment, L_1, L_2, the virtual position of the target point (x_Φ,y_Φ), and the virtual position of the motor (x_B,y_B). The unknown values are the angles b_1, b_2, and the length $\overline{B\Phi}$. This methodology aims to solve the real angle b°. The process of obtaining the distance of $\overline{B\Phi}$, this will be used in all future calculations. To calculate the x-coordinate and y-coordinate distance between Φ and B a

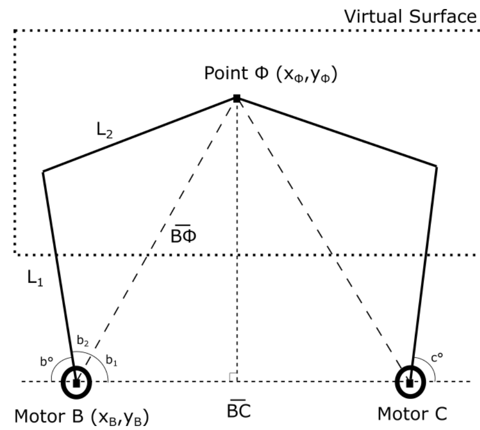

Figure 2: The mathematics behind the geometry of a two-segment arm

trivial 2-Dimensional Cartesian space subtraction is used:

$$\begin{array}{c} f(dx) = |x_\Phi - x_B| \\ f(dy) = |y_\Phi - y_B| \end{array} \quad (1)$$

Using 1, the distance between Φ and B can be determined by using the Euclidean distance formula:

$$f(\overline{B\Phi}) = \sqrt{dx^2 + dy^2} \quad (2)$$

Using 1, the arc-tangent formula will provide the angle between the opposite dy and adjacent dx:

$$f(b_1) = \begin{cases} arctan\left(\frac{dy}{dx}\right) & \text{if} \quad dx > 0, \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \quad (3)$$

Using 2, the inverse-cosine rule can be applied inside the triangle $\Delta L_1 \overline{B\Phi} L_2$ to determine the angle between L_1 and $\overline{B\Phi}$:

$$f(b_2) = \begin{cases} cos^{-1}\left(\frac{{L_1}^2 + \overline{B\Phi}^2 - {L_2}^2}{2 \times L_1 \times \overline{B\Phi}}\right) & \text{if} \quad L_1 \times \overline{B\Phi} > 0, \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \quad (4)$$

The final step uses 3 and 4, and the equation of a straight line to find the real angle of b°:

$$f(b^\circ) = \quad 180^\circ - (b_1 + b_2) \quad \text{for} \quad 0^\circ \leq b_1 + b_2 \leq 180^\circ \quad (5)$$

With b° obtained, it is possible to mirror the same methodology to determine the real angle of c°.

4.2 The three resulting robots

Throughout the construction, several design iterations of the LEGO Plot Clock outlined the components strengths and weaknesses. The three main components of the project were constructed – namely the physical robot and the software that controls it and the image process component. The design changes made during the development cycle and how these changes coincided with the rapid-prototype design principle. During the initial planning stages it was decided that the first and immediate milestone of the Plot Clock was to get a working physical robot as quickly as possible. Since the developers have no experience in robotics, robot programming, or mechanical engineering whatsoever, a week of practical experimentation was dedicated to allow time for getting familiar with the each component. Since the interactive servos are a key component to the project, they received a thorough examination. The first step was a trivial one, namely to connect one of the motors to port A on NXT Intelligent Brick and experiment with rotating it forwards and backwards.

Initial testing found a number of unforeseen problems with the performance of the Plot Clock Alpha. A trivial unit test to write out the digits 0 to 9 would take an extended amount of time. The written result also saw areas of concern that required addressing. These concerns are listed below:

1. **Movement Jitter** - The movement jitter, or noise, generated by the mechanical arms is predominantly caused by the friction between the marker tip and the drawing surface. The resulting drag would smear the line and in some cases prevent the tip from reaching an intended physical point – leaving gaps in the incomplete drawing.

2. **Pen distance to surface** - After several experiments had been conducted, it was found that the proximity of the pen tip to drawing surface had a significant effect on the quality of the written result. The smaller the distance between tip and surface, the less offset was caused.

3. **Angle Translation Inaccuracy** - The most concerning problem identified with the Alpha design was the marginal error of the NXT interactive servo. The [11] documentation explains that the movement functions are subjective to a variation of ±1 Tachocount inaccuracy when stopping. However, resulting start-stop tests found this to be closer to ±2.

To progress forward in the development cycle required several design changes to reduce the problems addressed in the previous subsection. These changes included replacing friction joints with non friction, raising the drawing surface so as to make it closer to the drawing tip, and adding more weight to the drawing arms. However, it was evident that a complete redesign was required because the inaccuracy of motors could not directly be resolved.

4.3 Image Processing software

To extrapolate the time from an analogue clock, a sequence of image processing techniques are used with OpenCV and NumPy in a Python project class. This software was intended to be as light-weight as possible, so the input stream was designed specifically for one type of clock face in fair room lighting conditions. The clock face chosen is a freely available Android application called *Simple Analog Clock version 3.5.3* [13], on the Google Play store[2]. With this, the program will have a clear target to capture. Additionally, the system clock is also displayed in the top-right corner of each screen capture – a useful metric to have when evaluating the test data. To assist in contextualizing the image processing logic flow, a list of pseudocode with each step partially explained is as follows:

1. Load web-cam feed from Microsoft LifeCam HD-3000.
2. Blur and threshold image to remove noise.
3. Find clock-face in image using template matching.
4. Find the 12h00 dot and 06h00 dot using template matching.
5. Determine base line between 12h00 dot and 16h00 dot.
6. Apply overlay on clock face to mask noise.
7. Find clock hands using Canny-edge detection.
8. Convert angle between baseline and each hand to time.
9. Write time to buffer file.

[2]Simple Analog Clock – goo.gl/sOZTTr

4.4 Results Achieved

Throughout the production life-cycle, attentive alterations were made to adhere to the original concept of Heberlein's open-source Plot Clock kit using the LEGO Mindstorms NXT kit. The first conspicuous feature is the initial topological design of the physical robot. Plot Clock Alpha shared near identical mechanical similarity with Heberlein's design: two upwards facing servos, the four-segment mechanical arm, the lifting servo situated between the writing surface and the center of mass, these are all matched design paradigm. However, the downside of using the LEGO as chassis material became apparent with the increase in overall size of the mechanism.

A number of possible factors for the popularity of Heberlein's robot could include its compact design, smooth transitional movement, and its effective, albeit bespoke purpose. The Plot Clock Theta does not share any of these traits whatsoever. The physical robot is significantly larger than the original Plot Clock, it requires more setup and overhead before it can be used to plot a single point, and has been designed to suit a platform role that could be used for future systems. These observations reflect back to the key question of if the system is a working solution. Since the robot can autonomously write the time, it meets this criteria. But, it definitely under-performs in terms of functionality in comparison to the original Plot Clock.

To address the criterion, a running catalog of written results was recorded from the start of the production lifecycle to track the overall refinement of the system over time. The output was recorded on standard A4 copier paper and scanned in to be evaluated. These scanned results will be compared to the desired written digital format. From this, it is evident that there has been a clear increase in accuracy over the three iterations of the Plot Clock. The Alpha version's results had no structure to the shape or formatting. This was improved with the Beta version, as lines started to straighten during pen strokes – the drastic change in drawing arm stability contributed to this resolution. An evolution of the robot's drawing capabilities can be seen in Figures 3–5.

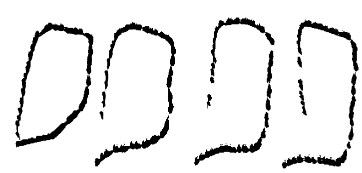

Figure 3: Plot Clock Alpha Result

Qualitative data was used to ascertain the overall success of the whole system and clear system limitations were outlined in each subsection. The ACR was analyzed and

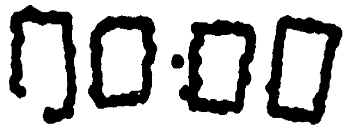

Figure 4: Plot Clock Alpha Result

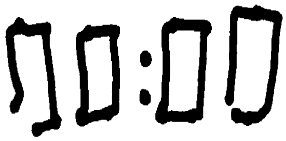

Figure 5: Plot Clock Theta Result

deemed a suitable addition to the project; the software was found to be suitably robust, and has a low margin for error when functioning inside ideal circumstances. Furthermore, each system layer was included in the system analysis, and a cross-examination with Heberlein's original Plot Clock was used a comparable system for context purposes.

5. CONCLUSIONS

The aim of this project was to successfully create a modular robotics system using the rapid prototyping design philosophy and incorporated with an external field of computer science knowledge. The problem domain was segmented into three interlinked subsystems, each posing its own set of challenges that required innovative approaches to be solved. A thorough investigation of the original Plot Clock, an open-source hobby kit designed by Heberlein, [8], lead to development of a modular LEGO Mindstorms NXT robot which underwent three system design life-cycles as evident by its three system versions. The chosen integrated field of knowledge was a real-time analog clock reader that was capable of extrapolating the time from a live web-cam video feed, generate a standardized time format, and then communicate the result to the running Plot Clock.

It can be concluded that the approach used to develop this project was successful in recreating the original Plot Clock in the LEGO Mindstorms NXT system with a modular driving software back-end and real-time image process component input. The system is robust enough to be used in condition and is programmed with a modular design philosophy in mind – enabling possible future extensions to simply pipe their own standardized input safely into the driving software. Several countermeasures have been coded into the robot to prevent hazardous code from executing. However, the system is not perfect, and several limitations have been explained.

5.1 Future Work

A practical achievement for this research is the implementation of a fully functioning mechanical plotting robot. Various future uses and extensions have been considered adequate additions for this project – but with an open-ended output such as the Plot Clock, the possibilities may only be bound by the physical limitation of the mechanical parts. A few possible extensions are outlined below:

1. **Further refinement** - The accuracy of the drawing could use additional investigation and refinement to fully replicate the smooth output produced by Heberlein's Plot Clock.
2. **Real-time interaction** - One interesting extension would be the integration a graphics tablet and stylus to interface directly with robot's mechanical drawing arm. This would be a challenging matrix manipulation and feedback domain problem - and would certainly be valuable as an Honours level project.
3. **Automatize with RPi** - One of the original goals of the system was to execute the running code on a portable Raspberry Pi, allowing a degree of physical freedom to the entire system. However, with the lack of Windows support required by the NXT USB and Bluetooth drivers, the incorporation of a RPi was not feasible in this project's time frame.
4. **Mobile application integration** - The NXT Intelligent is capable of sending and receiving messages via Bluetooth, so a mobile app that interfaces directly with the robot's mechanical arm could feasibly generate movements based on remote commands sent to it. This would be an interesting addition to integrate with the modular design of the driving software.

6. REFERENCES

[1] Adept. Research Robots - Adept MobileRobots. Online. Available: http://www.mobilerobots.com/ResearchRobots.aspx., 2015. Accessed: 2015.05.27.

[2] Atmel Staff. Plotting time with an Arduino Uno. Online. Available: http://blog.atmel.com/2014/02/20/plotting-time-with-an-arduino-uno/, 2014. Accessed: 2015.03.01.

[3] D. Blank, L. Meeden, and D. Kumar. Python Robotics: An Environment for Exploring Robotics Beyond LEGOs. In *In Proceedings of the SIGCSE Conference*, pages 317–321. ACM Press, 2003.

[4] Boise State University. Raspberry Pi - Boise State University. Online. Available: http://coen.boisestate.edu/ece/raspberry-pi/, 2013. Accessed: 2015.05.29.

[5] D. Chheda, D. Darde, and S. Chitalia. Smart Projectors using Remote Controlled Raspberry Pi. *International Journal of Computer Applications*, 82(16):6–11, 2013.

[6] D. Cook. *Robot Building for Beginners*. Technology in Action. Apress, January 2002.

[7] J. Heberlein. PlotClock by joo - Thingiverse. Online. Available: http://www.thingiverse.com/thing:248009/, 2014. Accessed: 2015.05.26.

[8] J. Heberlein. PlotClock source files - GitHub. Online. Available: https://github.com/9a/plotclock/blob/master/plotclock.ino, 2014. Accessed: 2015.05.26.

[9] F. Klassner and S. D. Anderson. Lego MindStorms: Not just for K-12 anymore. *IEEE Robotics & Automation Magazine*, 10(2):12–18, 2003.

[10] LEGO. WhatŠs NXT? LEGO Group Unveils LEGO® MINDSTORMS NXT Robotics Toolset at Consumer Electronics Show. Press Release. Retrieved from: http://www.lego.com/eng/info/default.asp?page=pressdetail&contentid=17278&countrycode=2057&yearcode=&archive=false, January 2006.

[11] leJOS. Controlling the Motors. Online. Available: http://www.lejos.org/nxt/nxj/tutorial/MotorTutorial/ControllingMotors.htm, 2015. Accessed: 2015.10.31.

[12] M. Milosevic, A. Dzhagaryan, E. Jovanov, and A. Milenković. An environment for automated power measurements on mobile computing platforms. In *Proceedings of the 51st ACM Southeast Conference*, page 19. ACM, 2013.

[13] T. Ono. Simple Analog Clock [Widget]. Online. Available: https://play.google.com/store/apps/details?id=net.moemoe.tomorrow.myAnalogClock2, 2015. Accessed: 2015.10.31.

[14] The Handy Board. MIT Handyboard and Super Cricket platform controllers. Online. Available: http://www.handyboard.com/., 2015. Accessed: 2015.05.27.

[15] S. Watkiss. Design and build a Raspberry Pi robot. Online. Available: http://www.penguintutor.com/electronics/robot/rubyrobot-detailedguide.pdf, 2015. Accessed: 2015.05.28.