KUKA YouBot Revival and System Development Project

Documentation of progress through project, in requirements of Final Project in DSD

Author: Aditya Agrawal

Advisor: assistant professor Tomasz Kucner

Supervisor: senior university lecturer Salu Ylirisku

ELEC-C0302 Final Project in Digital Systems and Design
Aalto University

May 14, 2025

Contents

1 Project Overview

The initial goal was to conduct a comprehensive revival of the robot and enable it to autonomously navigate a static environment using a low-cost camera. However, the goals shifted towards a more free-form exploration of the robot's capabilities and limitations. This document aims to document my progress and learnings, the challenges faced, and the conclusions drawn upon completion.

1.1 Motivation

The motivation for this project is to document the current state of the YouBot, repair and revive it to a usable state, and explore its current capabilities and limitations, as well as the potential for future applications.

Alongside completing this project, I personally wished to gain a deeper appreciation for the worlds of navigation and perception within to robotics, as well as understand the interfaces used to operate and customize this robot. This includes understanding the ROS ecosystem, open-source software like libfreenect [cite], the various drivers and wrappers used to interface with the robot through Ubuntu, the bash scripting language, the networking protocols used to communicate between the onboard computer, the sensors as well as external computers, and the hardware components of the robot itself.

There were significant learnings made in terms of project management and planning as well, as this project was a significant departure from any other course I have taken in the past. The open-ended nature of the project and the lack of clear goals made it difficult to stay motivated at times, but it also allowed for a more free-form exploration of the robot's capabilities and limitations.

2 Initial state of YouBot

This section aims to document the KUKA YouBot's state as initially received. This includes the actual robot base itself, the miscellaneous peripherals and accessories that were included, the onboard computer and its software, etc., as well as the documentation currently available for the YouBot, online and otherwise.

The KUKA YouBot was procured by the university in 2014, and has not been in use since 2019. At that time, the robot was considered to be quite antiquated, and the software and hardware were already outdated. Furthermore, the relevant websites hosting documentation and information are no longer available. This situation has only worsened over the years, and the starting point for this project documentation-wise was an user manual uploaded on a third-party website marked as a draft version from 2013.

2.1 About the Hardware

The YouBot is a mobile robotic platform developed by German automation company KUKA. First introduced in the early 2010s [cite], it was primarily designed for research and educational purposes in the field of robotics. To further this purpose, a significant portion of software used on the YouBot is open-source and available on GitHub [cite].

The YouBot typically consists of two main parts: a mobile base and a robotic arm. The mobile base is equipped with four omnidirectional mecanum wheels and motors for movement, alongside an onboard computer for processing and control. This onboard computer uses Ubuntu and ROS1 with conveninently provided drivers and wrappers, allowing for smooth software integration as well. The robotic arm has 5 degrees of freedom (DOF) and a two-finger gripper [cite], allowing it to potentially perform a variety of tasks through the onboard computer. Since our project involves only the mobile base, we will not be discussing the arm in detail.

The YouBot was also equipped with a variety of sensors when I received it, including a Kinect v1 camera [cite] and a functioning Hokuyo URG-04LX laser rangefinder [cite]. The Kinect camera module contains a infrared depth camera and RGB camera. This combined with its high availability and low cost made it a popular choice for navigation and perception tasks in robotics. The Hokuyo URG-04LX is a 2D laser rangefinder that was commonly used in robotics research as well, due to its compact design and high-resolution distance measurements.

This robot and the attached sensor modules are thus particularly well-suited for research within mobile robotics, particularly those pertaining to navigation and perception, as well as human-robot interaction.

2.2 Robot Base Overview

initial inspection

The robot base features a 24V power input via a 3-pin XLR connector, accessed through the top of the robot. In addition to this, there are two 24V 3-pin XLR output ports, indeded to power external sensors or other components such as robotic arms. On this top panel, one can also find two EtherCAT ports for consistent real-time communication with

motion-oriented robotics (i.e. the robotic arm), as well as a standard Ethernet port for wired interfacing with an external computer or network.

Mobility is provided by four omnidirectional mecanum wheels and relevant motors. This allows for relatively smooth and precise movement through a combination of linear and rotational motion. The motors are controlled via the onboard computer, which can send and recieve data from the motors and provide relevant feedback.

The onboard computer includes 6 USB2.0 ports, which can be used to connect various peripherals, such as a keyboard, mouse, USB, wireless adapter, etc. Additionally, there is a VGA port for connecting a monitor. [1]

The attached top panel is a sensor and mounting plate, designed to allow for the convenient attachment of various sensors and accessories. [cite https://web.archive.org/web/20160613151621/http://store.com/accessories/mounting-and-sensor-plate]

The robot base is also equipped with a slot for a sealed lead-acid battery (SLA), which can be used to power the robot when not connected to a wall outlet. The battery can connect to the robot via a 4-pin XLR connector.

2.3 Additional Hardware Components

DESCRIBE THE INITIAL INSPECTION IN FULL HERE! TALK ABOUT THE PILLAR STUFF AND NUTS AND WHATNOT AS WELL!!!!

2.4 Onboard Computer Overview

The computer runs [this software] and ROS version []. I wasn't able to immediately boot up any tests to ensure if the robot itself was working at any stage in time.

what ports are there? what software is on there? ubuntu? initial setup? ROS version? software? The onboard computer is an [] with an [].

It was running Ubuntu 12.04 LTS with ROS Hydro, which is a decade-old version of the operating system and the robot operating system. This version of Ubuntu is no longer supported, and the software is outdated and not compatible with most modern software and libraries. This will be detailed in a future section.

Alongside the OS and ROS, the computer also had various drivers and wrappers installed to enable communication with the robot's motors and sensors. This could be done directly through C++ programs, or through the use of pre-made ROS packages. We have elected to use the latter for the purposes of developing a system on the YouBot, to allow for a seamless integration with the ROS ecosystem in the future, alongside a level of standardization and ease of use.

2.5 Battery

The mobile robot base allows for a 24V power supply through a 4-pin XLR connector. It could also be operated without using a wired cable to the wall, by using a lead-acid battery originally which had a capacity of 5Ah and gave the YouBot an approximate runtime of 90 minutes. This lead-acid battery is [dimensions here]. This battery is also connected through a 4-pin XLR connector. While not a standard connector for power supply (4-pin

XLR connectors are typically used for audio equipment), it is very robust and secure for the purposes of a mobile robot.

However, the SLA batteries that were initially included with the YouBot were completely unusable. The lab engineer (Vesa Korhonen) had also put together a makeshift SLA battery in 2019, which had also degraded severely.

As such, either the original batteries had to be replaced, or the robot had to be continuously used with a wired connection to the wall. This would have been a significant limitation to the project had we not been able to create a makeshift replacement battery which will be discussed in section 4.4.

2.6 Sensors

Alongside the robot base and various hardware components, the YouBot was also equipped with a variety of sensors. These included a Kinect v1 camera and two Hokuyo URG-04LX laser rangefinders (, one of which had a broken mini-USB port, rendering it unusable). These sensors are commonly used in robotics research and are well-suited for navigation and perception tasks.

2.6.1 Kinect

The Kinect v1 camera is a depth and RGB camera that was originally designed and sold in tandem with the XBOX 360 to enable motion tracking and gaming. Due to its low cost, high availability and ease of use, it had indirectly become a popular choice for robotics research as well.

This Kinect camera module can be used through the use of open-source software such as libfreenect [cite] to access the data from the various sensors on the module. Furthermore, this information could be processed using the OpenCV library and then used to enable the autonomous navigation as previously described.

2.6.2 Hokuyo URG-04LX Laser Rangefinder

The Hokuyo URG-04LX is a 2D laser rangefinder that is commonly used in robotics research. It is a compact and lightweight sensor that provides high-resolution distance measurements in a 240-degree field of view. The URG-04LX is capable of measuring distances up to 4 meters with an accuracy of +/- 10 mm. It communicates with the onboard computer using a serial interface, making it easy to integrate into existing systems.

2.7 Documentation

The documentation for the KUKA YouBot is sparse and unavailable for the most part. KUKA has removed most references to the product from their website, and the original youbot-store.com website is no longer available. Furthermore, the only available user manual for the YouBot online is a decade-old PDF file that was uploaded by a third-party user, and is marked as a draft version from 2013.

Thankfully, the YouBot was marketed as an open-source platform, and the drivers as well as various parts of software are available on GitHub. Furthermore, we were able to find some more information about the YouBot through the Wayback Machine, which had

archived parts of the original website. While this archive is not complete by any means and is very tedious to comb through, it provides us with a powerful tool to piece together data about the YouBot and its use cases during its hayday.

We've also taken the liberty of citing some YouBot related projects here.

flow chart? how is the software and the hardware interacting with each other? pictures, diagrams, etc.

3 Project Goals

this section im not sure if it needs to be here anymore. discuss with advisor. The initial goal was to "revive the robot to a usable state and enable it to autonomously navigate a static environment using Visual Teach & Repeat [VT&R]. Add in more details from the initial project plan here.

learning goals technical goals: deployment, measurement of the quality of the odometry

problem of the phrasing

the set of goals have been formulated deviation from goals, which has been formulated from experimenting and identifying the limitations of the robot.

expansion of the goals setup of system

software, metholodogies

3.1 Navigation

What is VT&R?

3.2 Required Hardware

So for this we were expecting the robot base to work, the battery to work, and the kinect camera to work. And also I used another ubuntu machine to ensure the computing wasnt handled on the ancient piece of hardware.

3.3 Intended Setup

Our intended setup was to use the YouBot as a mobile base with the Kinect camera mounted on top of it. The onboard computer of the YouBot would only be used to control the motors and communicate with the sensors. The actual bulk of the processing would be done on a seperate computer that runs ROS2 and enables the necessary navigation and perception stacks.

3.4 Teach & Repeat

The initial prospect for this robot was to enable it to autonomously navigate using the above-mentioned sensors. Teach & Repeat (T&R) was considered to be an ideal candidate to allow for the same.

Teach & Repeat (T&R) [cite] is a two-phase robotic navigation method where a robot is first "taught" a path via human guidance or pre-recoded data. The sensor data captured during this phase can then be used to allow the robot to autonomously "repeat" the path later on, even in different environments.

T&R only requires a single camera for a basic implementation. It is a relatively simple method to implement through the use of open-source software and libraries, such as OpenCV and ROS. Furthermore, it is robust to changes in the environment and can dynamically correct errors through the use of visual odometry. As such, it has been a very

popular research topic within the field of robotics, particularly in the context of mobile robots [cite]. Much of the work done on extending T&R aims to improve this robustness and scalability for a variety of sensors and use-cases.

Thus, it seemed ideal for our project considering the age of the YouBot, the sensors available, and my lack of experience within the field of robotics.

4 System Development

4.1 Booting up the YouBot

describe the booting up process here, what ports to use, how to connect to the onboard computer, etc.

furthermore describe issues with software that needed to be fixed, such as the bashrc file, the networking, etc.

The initial inspection was carried out as described in section 2. Since the onboard battery was completely dead, I connected the YouBot to a 24V power supply and booted it up. On long pressing the power button, the screen flashed on and I could see the voltage input for the robot, alongside options to turn the PC and motors on/off seperately.

After connecting a VGA monitor and keyboard and mouse, I was able to boot up the onboard PC and use it as any other computer. The wireless adapter connected to this onboard computer was also functional, and I was able to connect to the internet. This meant I could theoretically SSH into the computer from my personal laptop, but I was unable to do so at that time due to a lack of a static IP address.

I was able to see the following software installed on the onboard computer: insert ROS version here and discuss the github repos that were already installed here.

There's a lot of files from at least 6 years ago on this, and I believe a significant portion are from like 2013 or 14. Two folders that look especially important are ros_stacks and youbot_driver. One file of high importance is in the desktop: the Kuka YouBot User Manual. I took a copy of that to my personal computer using a thumbdrive.

discuss powering up, initial software inspection, static IP issue

4.2 Running original demos

describe fixing network issue, and hten running the original demos.

4.3 Running ROS interface

describe the process of inspecting the installed software, the drivers and wrappers, and running the original demos. rqt, rviz, writing own packages, catkin, the various topics and nodes, etc. Finding the packages running these packages

4.4 ROS1-ROS2 Bridge

One of the first tasks I decided to embark on after getting the robot up and running was to set up the ROS1-ROS2 bridge. This was the most important link within the system, as it would allow the robot to seamless access the ROS2 navigation stack and enable it to access the computational resources of a more recent computer. However, the version of ROS1 on the robot is Hydro, which is more than a decade old and predates ROS2 development, i.e. it is not compatible with a ROS1-ROS2 bridge.

There are thus a handful of options to consider: One would be to upgrade the ROS1 version of the robot to a version that supports the bridge, such as Melodic or Noetic. However, this would require a significant amount of work and may cause compatibility issues with the outdated software on the robot. Another option would be to use a different computer with a more recent version of ROS1 and use it as a middleware between the robot and the ROS2 navigation stack. This would allow for a more seamless integration of the two systems, but would require additional hardware and software setup and create an unneccesarily complex system.

4.5 Writing Custom Programs

4.6 Experimental Setup for Movement and Odometry

here, ill discuss the experimental setup for testing movement and odometry. this should include the two experiments of moving linearly and rotating, and discuss how error is calculated. probably discuss negative feedback and PID control as well, random, percentage, consistent error, etc.

Considering the drift and error noticed whilst running my own programs, an experiment was set up for testing the movement and odometry of the YouBot. The error seems to be a combination of the robot's own motor control, as well as the odometry. It would be interesting to see if this error is systematic, random, and how it may be corrected in both cases.

different speeds? proportional or fixed error? does the odometry also drift?

20 times per experiment

4.7 Battery Replacement

draft 1, 11.05 12pm
Here are some good web pages about the Slealed Lead Acid (SLA) batteries: https://batterymasters.co.uk/ https://www.power-sonic.com/blog/how-to-charge-a-lead-acid-battery/ and https://www.powerstream.co

As previously mentioned, the YouBot came with three 24V SLA batteries. Two of these were the original batteries included with the YouBot, and one was a makeshift battery put together by Vesa in 2019. All three batteries were unusable. This section thus documents the technology behind the SLA batteries, the testing of the original batteries, replacing and testing a new battery, and future battery options.

4.7.1 SLA Battery Overview

Sealed lead-acid (SLA) batteries operate based on a reversible chemical reaction between lead plates, lead dioxide and sulfuric acid electrolyte. When the battery discharges, the substances react to form lead sultfate and water, releasing electrical energy. During charging, this reaction is reversed.

These batteries are quite simple, robust, inexpensive and safe to use, indicating the reason for use in this scenario. However, they require some maintainence, and should undergo full discharge and charge cycles regularly to keep them in a good state. If they are not used regularly, they can suffer from sulphation, where lead sulfate crystals gradually form on the plates of the battery. This process is irreversible and permanently impairs the battery's capabilities. Cheaper SLA batteries are more prone to this issue due to lower quality materials, and one may only expect a maximum lifespan of 3-5 years from them.

SLA batteries are furthermore sealed and contain one-way valves to prevent internal pressure buildup due to production of hydrogen and oxygen gas. Normally these gases recombine back into water, but overcharging can force gas release, leading in gradual water loss.

The recommended voltage for charging SLA batteries is 2.3 volts per cell (2V), or 13.8V for a 6-cell (12V) battery, or 27.6V for a 12-cell (24V) battery. Charging at a lower voltage (i.e. 2V per cell) will not fill up the battery completely, and increase the risk of sulphation, since the lead sulfate crystals will not be fully converted back into active materials.

4.7.2 Testing original batteries

To confirm the degradation of the original batteries, we decided to test them using a multimeter. The two original batteries were completely dead, and did not show any voltage when connected to the multimeter. The makeshift battery showed some voltage, but it was far too low to be usable out the gate. This battery was put together using two 12V SLA batteries, where each battery was individually connected to the 4-pin XLR connector. The YouBot internally connects them in series to create a 24V battery.

Vesa attempted to revive this battery through desulphation, where a higher-than-recommended voltage (in this case, 30V) is applied to the battery to force current through the hardened sulphate layers and dissolve them.

This seemed initially promising: the battery was accepting a charge and its voltage was increasing. However, this was only temporary, as the battery quickly lost voltage again after

charging, suggesting that the sulphation was very severe. While some surface conductivity was perhaps restored, the effective area of the electrode plates was still very small, resulting in a very low capacity. As such, the battery was not usable for our purposes.

4.7.3 Replacing batteries

maybe here we can link to the specific batteries used? the dimensions would be neat as well...

We decided to remake the makeshift battery using two new 12V SLA batteries of the same dimensions. While the previous batteries were Bitelma batteries, we bought some from Leader this time. These batteries were 12V 5.4Ah batteries. We tested them using a multimeter and used a car charger to charge them overnight.

The old batteries were removed from their casing and all relevant connections and pieces to structure the battery were removed. The new batteries were then appropriately connected to the 4-pin XLR connector maybe mention the pinout here?, and the structural pieces were reattached with tape. After putting the casing back on, we were able to connect the new battery to the YouBot and power it on. The YouBot detects the two batteries and shows their individual voltages, confirming that the battery works. While it's not hermetically sealed, it works well for the purposes of this project.

4.7.4 Future Battery Options

While we have replaced the SLA batteries with a makeshift one at a rather inexpensive cost, these batteries will not last long and will be prone to the same issues as the original batteries.

An ideal candidate for a replacement battery techonology would be lithium-ion-phosphate (LiFePO4) batteries. These batteries are more expensive, but have multiple advantages over SLA batteries. They have longer lifespans, higher energy density, lighter weight, and are less prone to degradation. However, they would also need an integrated battery management system (BMS) to ensure safe operation, and potentially some custom hardware to fit the dimensions of the YouBot's battery compartment.

4.8 Sensors

4.8.1 Kinect Camera and libfreenect

The Kinectv1 camera is a depth and RGB camera that was originall designed and sold in tandem with the XBOX 360 to enable motion tracking and gaming. Due to its low cost, high availability and ease of use, it has indirectly become a popular choice for robotics research as well.

We can use open-source software such as libfreenect to access the data from the various sensors on the module.

discuss the libfreenect library, how to install it, and how it could be used within ROS.

4.8.2 Hokuyo URG-04LX Laser Rangefinder

The Hokuyo URG-04LX is a 2D laser rangefinder that is commonly used in robotics research. It is a compact and lightweight sensor that provides high-resolution distance measurements in a 240-degree field of view. The URG-04LX is capable of measuring distances up to 4 meters with an accuracy of +/- 10 mm. It communicates with the onboard computer using a serial interface, making it easy to integrate into existing systems.

discuss installing the drivers and whatnot, and how to use it within ROS and rqt.

5 Future Applications

In this section we will discuss a potential future application of the KUKA YouBot, in the context of the T&R method of navigation. The T&R method is a form of visual SLAM, where the robot can be "taught" a path through a static environment using a camera. The robot then uses this information to enable it to repeat the path movement autonomously through key frames it has saved from the environment. This method is simple yet effective in its use cases and can be used in a variety of applications, such as industrial automation, home robotics, service, etc.

t&r is not the best implementation of the youbot

platform and real world application

why im a bit skeptical about trying to implement t&r in the context of assistive platform trying to present as an application as a playform that can later on be used to execute repetitive task based on the experience of a platform with

speculative : presenting it in terms of real-world applications

flexible intra-logistics human awareness and flexibility seamless human-robot existence

While the YouBot holds the potential capabilities for undertaking Teach and Repeat, its abilities as a mobile industrial robot base would be better suited for real-world applications within closed environments as an assistive platform.

This robot base would be most useful in an assisstie sense, where it assists humans in completing rather basic and repetitive tasks, like fetching beer.

6 Reflection

draft 1, 11.05 6pm

In this section, I will reflect on the outcomes of this project. This will include an overview of project management practices and planning, alongside my learnings and personal feelings about the project.

6.1 Project Evolution

As disccused previously, the initial goal of this project was much more concrete and ambitious, with an ultimate aim of enabling Teach and Repeat on a decade-old robot whilst using something like ROS2 Jazzy. This was simply not feasible with the outdated software and hardware, lack of documentation, my own lack of experience, and the overall time and resource constraints for completing this project. Furthermore, this project was a significant different experience from any other course I have taken in the past.

My initial approach to project management was very linear and straighforward, with the understanding that there would be no hiccups or issues along the way. I had the following time-table in mind: maybe insert a table showcasing the plan here? While this was initially rewarding as I made progress in individual tasks (i.e. Kinect camera, booting up computer, static IP, etc.), momentum was quickly lost when it came to integrating these tasks into a cohesive system, through the use of the ROS1-ROS2 bridge.

6.2 AGILE

The new mindset I undertook was inspired by the AGILE project management methodology, which takes a much more flexible and iterative approach to project management. While the long-term goals are still important, the focus is on short-term goals and iterations. This allows for a more flexible approach, where the ending goal can be adjusted based on constraints, feedback, and progress.

While this also indirectly means that the project may not be concretely completed, it led to a more flexible exploration of the robot's capabilities and limitations. It furthermore relieved me of the pressure of having to complete a specific (and steep) goal, and gave me breathing space to play around with the available components more freely. This was much more rewarding and enjoyable.

Undertaking this mindset required time and patience, where I had to learn to accept that not everything would go according to plan. It furthermore required me to be more open to feedback, and actively focusing on short-term goals.

6.3 Final Thoughts

Overall, despite the various challenges faced, I am very happy with my progress and learning outcomes regarding this project. I was able to successfully revive the KUKA YouBot to a usable state, gain a deeper understanding of the ROS ecosystem, Linux, bash scripting, alongside the various hardware components of the robot. More importantly, I learned to manage projects with the understanding of personal and technical constraints. I'd like to thank my advisor for his provision of a different perspective on the project, explanations of everything and constant patience.

References

[1] "Youbot - Overview — github.com", https://github.com/youbot, [Accessed 23-04-2025] (page 3).

[2] I. Tsogias, "Kinematic Analysis of a KUKA YouBot", 2016. DOI: 10.26240/heal.ntua.12581. [Online]. Available: https://dspace.lib.ntua.gr/xmlui/handle/123456789/43358.

[3] "YouBot Store — web.archive.org", [Accessed 23-04-2025]. [Online]. Available: https://web.archive.org/web/20170318163736/http://www.youbot-store.com/developers/.

[4] "YouBot Driver — janpaulus.github.io", https://janpaulus.github.io/, [Accessed 23-04-2025].

A User Manual for the KUKA YouBot

A.1 How to turn on the YouBot

A.2 .bashrc recap

A.3 How to run original demos