Connected and Autonomous Robotic System

Project Workbook

By

Jay Sureshbhai Parsana

Prashant Shushilbhai Gandhi

Saumil Shah

Vatsal Makani

3rd October 2019

Advisor: Prof. Kaikai Liu

Chapter 1. Literature Search, State of the Art

Literature Search

Since the past decade research on autonomous systems like self-driving cars, robots, etc is on the rise [2] and the tech industry will continue to develop techniques and algorithms to achieve complete self-driving experience. Using such algorithms, "experiments have broadened their scope and autonomous driving for several kilometers and over long periods of time" have become commercialized and available to all [5]. In recent times, companies are using laser-based radar (LIDAR) as well as traditional radars in robotic systems in order to add the surrounding awareness capabilities in these systems. Vijay Subramanian, Thomas F. Burks & A.A. Arroyo (2006) explained the use of radar incorporated with computer vision in autonomous tractors for citrus grove applications [1]. The authors explain the implementation of the "PID control algorithm to control the tractor using the information from the machine vision system and laser radar", but the only drawback was latency communication from radar to the platform that processes the radar data [1]. Zhengang Li, Yong Xiong, Lei Zhou (2017) explains the working of the indoor wheelchair performing the autonomous exploration based on inexpensive RGB camera and ROS. The authors explain that the wheelchair rotary systems are controlled by the robot operating system in the host computer and RGB camera is used for depth perception, indoor path planning and obstacle avoidance [4]. The Arduino microcontroller is then used to receive the data from the camera and ROS and accordingly sets the PWM signals for the motors controlling the navigation of the wheelchair.

There are many applications where LIDAR proves advantageous over radars. LIDARs are used in applications that involve short-range detection and radars are used for long-range tracking. LIDARs can be used to create a larger 3D field of view and accurate range information which is suitable for obstacle detection as well as pedestrian recognition as explained by Heng Wang, Bin Wang, Bingbing Liu, Xiaoli Meng & Guanghong Yang (2016) [2]. The authors also explain the integration of LIDARs with IMU sensors and filters for implementing localization and pedestrian recognition [2].

Some of the airborne automated systems require navigation control based on IMU sensor units like accelerometers, magnetometer, gyroscope, etc. integrated along with location data. Jan Wendel, Oliver Meister, Christian Schlaile, Gert F. Trommer (2006) explain an integrated navigation system for an airborne robotic system that uses GPS and MEMS IMU sensors [3]. The authors propose the solutions for two scenarios: first when GPS data is available and second when GPS is lost [3]. Another replacement for GPS can be ultrasonic beacon which can be used for indoor positioning. Dongho Kang & Young-Jin Cha (2018) explains the use of ultrasonic beacon in the scenarios where GPS fails to work for a UAVs. The authors explain the application of a "deep convolutional neural network (CNN) for damage detection and a geotagging method for the localization of damage" [8]. "Localization, mapping, and path planning" are the most important algorithms used for any system to be autonomous. Localization, motion planning and path planning plays an important role in deciding the shortest path that any autonomous system can travel in minimum time [6].

If an autonomous system needs to monitor the surroundings keeping the functionalities such as localization, path planning, and mapping, it is mandatory to define two different architectures communicating via IoT. LoRA is another wide area network telecommunication system which is widely used for IoT Architecture [7]. LoRA provides higher bandwidth and is basically targeted for low power systems like embedded microcontrollers.

State-of-the-Art Summary

The higher demand for autonomous and IoT based systems in the tech industry has accelerated the development of new techniques to optimize the current state-of-the-art solutions. The objective of the project is to provide a platform that can be utilized for any autonomous applications in robots, cars, warehouses, hospitals, etc. We plan on minimizing the time for developing such autonomous system by providing a base platform on which many OEM's can integrate their hardware and use our API's to get a head-start for their system. This project consists of a low-level architecture that takes care of sensing, positioning, and obstacle avoidance and high-level architecture to compute localization, path planning, and generate control signals for low-level architecture. Architectures talk to each other using IoT framework such as LoRA or Wi-Fi. The communication between these architectures occurs at a very high speed, thus solving the problem of slow communication faced by Vijay, Thomas, and Arroyo [1].

The implementation of low-level architecture can be based on ST Microelectronics' STM32 platform and PX4 platform by px4.io. Due to the integration of many sensors in low-level architecture, the use of FreeRTOS will help to achieve task synchronization. Each task is dedicated to each sensor that executes in real-time by using various scheduling algorithms. STM32 has inbuilt support of CMSIS v2 to add real-time capabilities. We researched various technologies for establishing an IoT link such as the Verizon IoT module, Wi-Fi, LoRa. The Wi-Fi module in the low-level architecture enables the system to send sensor data to cloud services like AWS, Google IoT, Microsoft Azure, etc. Verizon IoT combines modules, network services, and the ThingSpace management platform to support IoT applications with minimal annual fee which is ideal for any embedded application.

Decaware provides a DWM1004C module for tagging and tracking applications. DWM1004C offers high accuracy, real-time location capability with a 6.8Mbps data rate which is useful for tag applications that are like RFID. Another tracking module is TREK1000 which is a two-way ranging module for real-time positioning applications. This module works in 3 modes to enable tracking, geo-fencing and navigation schemes.

The high-level architecture uses Nvidia Jetson TX2 for complex calculations like localization, path planning, and motion planning. The high-level architecture is based on Linux. The high-level system does not guarantee real-time operations. The software platform explored for high-level architectures are Autoware.AI and traditional ROS. Autoware.AI is an open-source software that implements localization with the help of 3D maps and SLAM algorithms, detection based on LIDAR and cameras and prediction based on deep learning neural networks. ROS provides APIs with inherent support for localization, path planning, and robot control.

References

1. Subramanian, V., Burks, T. F., & Arroyo, A. A. (2006). Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. Computers and electronics in agriculture, 53(2), 130-143.

   ["This paper discusses the development of an autonomous guidance system for use in a citrus grove using machine vision and LIDAR"]

2. Wang, H., Wang, B., Liu, B., Meng, X., & Yang, G. (2017). Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. Robotics and Autonomous Systems, 88, 71-78.

   ["This paper studies the pedestrian recognition and tracking problem for autonomous vehicles using a 3D LiDAR, a classifier trained by SVM (Support Vector Machine) is used to recognize pedestrians"]

3. Wendel, J., Meister, O., Schlaile, C., & Trommer, G. F. (2006). An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter. Aerospace Science and Technology, 10(6), 527-533.

   ["This paper addresses the development of an integrated navigation system based on MEMS inertial sensors and GPS for a VTOL-MAV. These MAVs show an inherent instability that makes at least an automatic stabilization necessary"]

4. Li, Z., Xiong, Y., & Zhou, L. (2017, December). ROS-Based Indoor Autonomous Exploration and Navigation Wheelchair. In 2017 10th International Symposium on Computational Intelligence and Design (ISCID) (Vol. 2, pp. 132-135). IEEE.

   ["Aiming at the current situation of high cost, complicated construction and poor reusability of the autonomous navigation system, an indoor wheelchair autonomous exploration and navigation system with low cost and high reusability is realized in this paper"]

5. Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. IEEE Transactions on Intelligent Vehicles, 2(3), 194-220.

["This paper proposes a survey of the simultaneous localization and mapping field when considering the recent evolution of autonomous driving, present the limits of classical approaches for autonomous driving and review the methods where the identified challenges are solved"]

6. Fethi, D., Nemra, A., Louadj, K., & Hamerlain, M. (2018). Simultaneous localization, mapping, and path planning for unmanned vehicle using optimal control. Advances in Mechanical Engineering, 10(1), 1687814017736653.

   ["In this paper, investigation of the path planning problem of unmanned ground vehicle is based on optimal control theory and simultaneous localization and mapping"]

7. Augustin, A., Yi, J., Clausen, T., & Townsley, W. (2016). A study of LoRa: Long range & low power networks for the internet of things. Sensors, 16(9), 1466.

   ["This paper provides an overview of LoRa and an in-depth analysis of its functional components"]

8. Kang, D., & Cha, Y. J. (2018). Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging. Computer-Aided Civil and Infrastructure Engineering, 33(10), 885-902.

   ["This paper proposes an autonomous UAV method using ultrasonic beacons to replace the role of GPS, a deep convolutional neural network (CNN) for damage detection, and a geo-tagging method for the localization of damage"]

9. Grewal, H., Matthews, A., Tea, R., & George, K. (2017, March). Lidar-based autonomous wheelchair. In 2017 IEEE Sensors Applications Symposium (SAS) (pp. 1-6). IEEE.

   ["This paper seeks to mitigate these problems by presenting a wheelchair capable of autonomous navigation, with minimal directive from the user"]

Chapter 2. Project Justification

There has been an increasing research interest and demand in the autonomous self-driving systems and IoT connected devices for the past decade. Currently, in the field of robotics and autonomous self-driving systems, platforms such as Slamtec Slamware exist. These platforms do not provide complete end to end solution that can be deployed on a robotic or automotive system and thus cannot be used to quickly prototype a product. Thus, engineers need to research and test different hardware modules and their software compatibility which suits their application requirements. We thereby propose to solve this problem using a generalized connected and autonomous robotic platform.

The purpose of our project is to develop a foundation which can be used for various applications in medical, robotics, industrial and automotive domain and targets to reduce the time to market by providing a complete package of hardware and software solutions for autonomous systems. This robotic system could be deployed directly on systems like cars, wheelchairs, industrial robots, etc. to adapt to various autonomous applications. Our project is an amalgam of embedded systems, IoT, distributed sensor networks, deep learning applications, etc. Our project shall comprise of 2 levels of architecture where low-level architecture supporting the integration of various kinds of sensors like telemetry, environmental and proximity sensors with the microcontroller for monitoring the surrounding parameters like environmental or an obstacle in the proximity. The high-level architecture is a Linux based Nvidia Jetson TX2 which supports computation of complex frameworks responsible for making the robotic system autonomous. Our project shall also provide IoT/Cloud support for data storage and monitoring applications.

Chapter 3. Identify Baseline Approaches

After intensive literature review of implementations of various autonomous systems and sensor integration, the baseline approaches for low-level architecture and high-level architecture were decided. There are few software platforms available in the market which can implement localization, path planning and control for the high-level architecture. Autoware.AI provides inherent support for these algorithms but it is compatible only with heavy computing hardware like Intel Core processors, x86 architectures and Nvidia Xavier development kits. Since Nvidia Xavier has similar architecture to Nvidia Jetson TX2, it is possible to evaluate the Autoware's software platform performance in TX2. The traditional ROS provides native support for APIs used for complex algorithms in robotics.

As a baseline product, we would like to implement the one explained in the paper "LIDAR based Autonomous wheelchair" [9]. The authors explained the implementation of LIDAR to navigate the wheelchair. LIDAR provides the surrounding measurements and mapping around the wheelchair, sends the navigation information to microcontrollers and rotary system to move the wheelchair in the dedicated path to avoid any obstacles with the help of ROS. Our project shall implement a similar base architecture with additional features such as IoT support and other sensor information. The authors suggest the use of PC based x86 architecture to implement a high-level system for ROS used for localization and object detection. Whereas our project proposes to use ARM-based NVIDIA Jetson TX2 board with GPU providing low powered operation and universally deployable platform. We also will be using an ARM-based STM32 low-level platform which is scalable, mature and widely used in the industry instead of Arduino based platform underpowered platform.

For low-level architecture, we researched PX4 and STM32 platforms, but integration of FreeRTOS with the PX4 platform turns out to be a time-consuming task since PX4 APIs are not developed to provide real-time performance and STM32 provides inherent support for FreeRTOS which makes it suitable for low power, real-time applications. We have found few research papers regarding the autonomous systems integrated with GPS, IMU sensors, deep learning neural networks, radars and LIDARs for applications like pedestrian detection, citrus grove navigation, oceanography, etc. The author of one of the research papers explains the application of a beacon when the GPS data is lost, the application of LIDAR for obstacle detection as well as prediction using deep learning neural networks.

The communication between low-level STM32 platform and high-level Nvidia Jetson platform can take place using any one of the device protocols like UART, SPI or I2C. The cloud and IoT support at a low level is used for remote monitoring the sensor data such as motion readings, GPS readings, environmental parameters, etc.

Dependencies and Deliverables

Dependencies

Our platform will provide features like object detection, coordinates of system, localization and path planning. While implementing these features, we need to consider some of the dependencies:

1. Getting GPS coordinates in closed areas such as buildings, warehouses, hospitals, etc.

2. For object avoidance, LIDARS fail to detect smooth surfaces such as walls. LIDARS are also sensitive to excessive light, causing it to give false readings.

3. Object detection – Prepare, define and find dataset relevant to the applications of our project. Choose appropriate machine learning algorithms and models for the best accuracy.

4. Establish communication between STM32 and Jetson TX2. Detection of obstacles and co-ordination of that information to avoid it will be a challenge as both are independent.

5. Integration of obstacle data from the high-level and low-level system to make a coordinated action to avoid it.

6. Implement APIs that are simple to use and easy to understand as a product.

7. Design the system so that it can be compatible with any future modifications.

Deliverables

Our project's objective is to build a generic platform that can be used by different robotic systems. For low-level architecture we plan to interface sensors such as accelerometer, gyroscope, GPS, Ultra-wideband beacon, IoT integration, IMU sensors, proximity sensors, and radars. We will deploy multiple STM32 boards which will communicate with each other through a common bus. Some redundant controllers and sensors will also be added for the system to be fail safe.

A high-level system will comprise of a Nvidia Jetson TX2 board on ROS for processing data such as images from camera and point cloud from stereo camera. The system will compute path planning and motion planning algorithms using the data from camera and LIDAR. It will also be used to implement machine learning models and detect obstacles. The high-level architecture will generate some control signals and control the robotic system partially based on camera and LIDAR input.

Chapter 4. Project Architecture

The block diagram below shows a clear picture of what we will be trying to achieve and how various systems will communicate with each other to achieve autonomy:

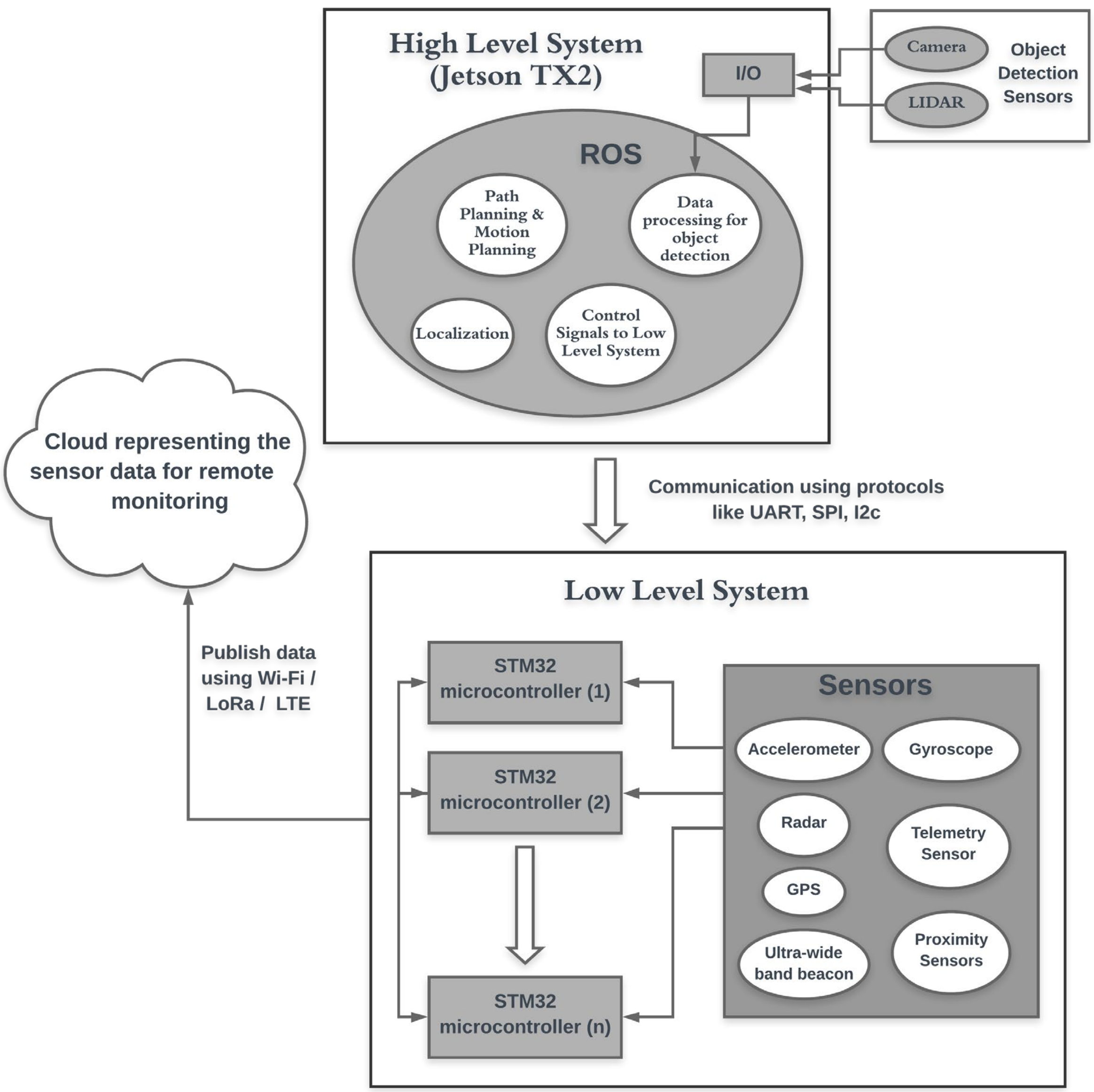

Figure 4.1 System block-diagram

The project architecture will be divided into two levels:

- Low-level architecture
    - The low-level architecture consists of an embedded system which is the STM32 IoT discovery kit with FreeRTOS.
    - IMU sensors will be used to get feedback while doing the motion planning.
    - Radar and proximity sensors will detect the obstacles and a control action will be taken to avoid them.
    - Ultra-wideband beacon will be used for the localization of the robot. It will connect with the nearest beacon for reference to find out where is it located.
    - Telemetry sensors will be used to monitor the speed of the robotic vehicle and take control actions like reducing and increasing the speed of the robot.
    - IoT integration for STM32 will be done using Wi-Fi or LoRa. It will be used to publish sensor data on the cloud for monitoring it remotely.
    - A low-level system will get control signals from a high-level system. This control signal will be used to avoid obstacles. The communication link will be established using SPI, I2c or UART protocol.
- High-level architecture
    - The high-level architecture will run on the **R**obot **O**perating **S**ystem (ROS). It is an OS that has several API's for path planning and motion planning based on the camera and LIDAR.
    - The data from the camera and LIDAR will be processed parallelly by utilizing the GPU.
    - Obstacle detection by implementing a deep learning model will also be achieved by using the camera feed.
    - A 3D point cloud from LIDAR will also assist in obstacle avoidance.
    - All this data and algorithm implementation will, in turn, be converted into control signals. These signals will be communicated to the low-level system so that it can avoid obstacles.

Chapter 5. Implementation Plan and Progress

- For low-level hardware, setup the programming and execution environment by installing and configuring the IDE, compilers, microcontroller's Software Development Kit (SDK).
- Shortlist and acquire the sensor to be used such as accelerometer, gyroscope, GPS by comparing available options as per the requirement of the project.
- Understand the available drivers, example and software stack for the low level and high-level platform and evaluate its use and performance is our application.
- Perform Hardware simulation and generate raw data to test the platform and verify our initial design and implementation procedure.
- Manage sensor collected data using well-defined tasks and co-ordinate their operation making it robust for future applications
- Study and analyze different IoT platforms and finalizing a cloud service that is scalable and easy to manage, making uploading senor data seamless.
- Transfer sensor data collected from low-level architecture to high-level architecture such that NVIDIA Jetson TX2 platform can analyze the sensor data pattern to take required actions for the robotic platform
- Get a deeper understanding of the working of the ROS platform and its supporting packages for deep learning applications, object detection, image segmentation.
- Learn the basics of CUDA programming and the OpenCV framework to optimize the performance of our deep learning and computer vision application on the NVIDIA Jetson TX2 platform.
- Develop a demo application for high-level architecture to showcase its capability and use case for deploying it in the robotic platform.
- Create a prototype of the entire low-level and high-level platform architecture of the project and deploy it to a practical application use case such as drones, autonomous vehicles, robots.

Chapter 6. Project Schedule

Sr. No.	Description	Team members (2 member team for each task)
1.	Finalize the details of the low-level and high-level architecture with the advisor.	Low-level architecture research – Jay and Saumil High-Level architecture research – Vatsal and Prashant
2.	Get started with the development environment and familiarize ourselves with hardware for low level and high-level architecture.	Setup the environment for STM32 and research about different sensors available in the market – Jay, and Saumil High-Level research: motion planning– Vatsal and Prashant
2.	Add FreeRTOS for the STM32 development board along with peripheral and software libraries to be used for Low-level architecture platform.	Acquire hardware and run all the software examples for sensor modules on STM32 board – Jay and Saumil High-Level research: path planning– Vatsal and Prashant
4.	Interface sensors such as accelerometer, gyroscope, GPS, Ultra-wideband beacon, LTE module, IMU sensors.	Develop drivers for GPS (parsing function), telemetry sensors and beacon – Jay and Saumil Develop drivers for Radar, ultrasonic sensor and IMU sensors (motion data capture)
5.	Create Separate tasks for each sensor and synchronize them to manage their data collection forming a reliable sensor network.	Collaborate the code and make the code work after collaborating effectively without any issues – All members
6.	As per the project requirement create PCBs and hardware enclosures for the low-level hardware architecture.	Collaborate and integrate all the sensors – Jay and Saumil Work on PCB, finalize it and collaborate for integration of sensors –

		Vatsal and Prashant
7.	Connect the low-level hardware system to the cloud enabling the Internet of Things by using services such as AWS IoT or Google Cloud to monitor and analyze the sensor data.	Publish all the sensor data on the cloud – Vatsal and Saumil Assemble the hardware and check the integration of hardware and sensors – Jay and Prashant
8.	Create an Ubuntu Image for Nvidia Jetson TX2 board and load the OS. Familiarize with the functioning and usage of Nvidia Jetson TX2.	Check the Camera on Jetson with the sample codes – Vatsal and Jay Check the LIDAR on Jetson with the sample codes – Saumil and Prashant
9.	Port ROS on Nvidia Jetson TX2 high-level platform and accompanying packages required for the creation of a Robotic System.	Understand ROS and explore the API's from ROS on Jetson – All team members
10.	Interface Camera and Lidar to Nvidia Jetson TX2 and receive data from them in a successful manner.	All team members
11.	Create 3D Maps for localization and deep learning applications for object detection, obstacle avoidance, and image segmentation.	Work on camera and object detection and deep learning models – Vatsal and Saumil Work on LIDAR for obstacle detection and develop a mechanism to create a signal for
12.	Deploy the low-level system and high-level system in a Robotic platform, Drone or autonomous vehicle to test and evaluate the performance of the system.	Work on path planning and motion planning and integrate the entire low level and high-level part of the project to make it as a complete package – All team members