Australasian Transport Research Forum 2023 Proceedings
Perth, Australia
Publication website: http://www.atrf.info

Developing an autonomous shuttle service

Kieran Quirke-Brown[1], Zhihui Lai[1], Xiangrui Kong[1], Tim Tan[1], Yuchen Du[1], Thomas Bräunl[1]

[1]The University of Western Australia, School of Engineering

Email for correspondence: kieran.quirke-brown@uwa.edu.au

Abstract

This paper describes the performance of the nUWAy autonomous shuttle bus that we established on the UWA campus as well as the initial performance on public roads in Amberton Beach. We acquired two secondhand shuttle buses for which we developed the complete software stack by combining open-source software packages with self-written software. The first shuttle bus conducts regular drives on the UWA campus, providing free transport for students and staff, while we are preparing the second shuttle bus for a public road trial at Amberton Beach. A firmware-level Lidar system acts as a safety curtain around the shuttles, stopping them if any obstacle comes too close to it. This is independent of the higher-level student-written software that runs the localization and path planning algorithms. The autonomous shuttle on campus provides a testbed for navigation algorithms, which we will expand for the autonomous shuttle operation on public roads.

1. Introduction

Current available public transport, particularly in places like Australia, allows users to travel to set locations within cities and towns using trains and buses. These systems adhere to timetables and locations where there is a large enough demand to make the system cost effective. This leaves many users with extended travel and wait times as they need to walk or ride to get to these set transport locations. While for some users this option is fine, there are many who avoid taking public transport due to the time and effort required to get to a train station or a bus stop. This is where autonomous shuttle buses become a viable solution. On-demand autonomous shuttle buses that are not restricted to fixed stops and timetables can improve users' public transport experience and reduce the overall travel time. The goal of the REV project is to design a smart and viable transport solution that can achieve this goal. Another application environment of autonomous shuttle buses are large campuses and plaza like airports where people need to move or carry baggage and equipment between buildings that are kilometers apart. Using on-demand autonomous services can assist users in transporting their luggage and arriving in a timely manner.

Since 2020, The Renewable Energy Vehicle (REV) Project at The University of Western Australia (UWA) has purchased two second-hand EasyMile shuttle buses with drive-by-wire and sensor systems installed, but without any software. The goals of this project are to build a complete software stack and improve current algorithms for autonomous driving in a closed environment as well as on public roads. In a second step, UWA Engineering will cooperate with the UWA Business School to evaluate user experience and expectations of autonomous shuttle services.

2. Vehicle hardware setup

The initial setup of the driving chassis came equipped with most of the required sensors and hardware in place. This included eight Lidar sensors, CAN bus motor controllers, cameras, wheel encoders and an Ethernet and CAN network. The existing GPS and IMU (inertial measurement unit) were replaced with an improved GPS+RTK (real-time kinematics) and IMU to improve performance. The hardware is described below.

Figure 1: nUWAy autonomous shuttle buses

The vehicle houses two 48V, 7.2kWh Lithium Iron Phosphate batteries with enough power to drive continuously during the day, depending on speed and peripheral usage, and a 12V lead acid starter battery. The original charging port on each bus was a non-standard socket that was replaced with a European standard IEC Type-2 AC charging port to be compatible with the existing charging stations. Power is distributed through the bus from the 48V batteries by two step-down converters providing three bus bars, 48V, 24V and 12V, respectively.

2.1 Motors and controllers

The bus houses a single drive motor with a gearbox connected to the front two wheels for driving. Each steering rack also has a separate steering motor which can be used individually (front or rear wheel steering) or together for an improved turning radius. The motor controllers receive CAN bus commands from a central PLC system. These PLC controls provide low-level safety through hardwired connections from the safety Lidars and various other sensors. During regular driving the bus will use regenerative braking to slow down and recover some of the mechanical energy and recharge the 48V batteries. A hydraulic brake is used to assist in braking and is controlled by a linear actuator. Each wheel is fitted with an electric caliper which clamps down on the wheels to stop any movement when the vehicle has come to a stop acting as a parking brake. An emergency brake system has been fitted to stop the vehicle in the case of power loss by applying force to the drive motor output axle, stopping it from moving.

2.2 GNSS + RTK

An SBG Ellipse-D dual antenna GNSS (Global Navigation Satellite System) was added for its high accuracy and onboard IMU features. To improve the accuracy of the GNSS reading, RTK (real-time kinematics) corrections are transmitted from a base station nearby over the Internet, using the highly accurate error offsets to correct the satellite data.

2.3 Lidars

We are using two main sensors for object detection, Lidars and cameras. The shuttle is fitted with eight Lidars: four Sick LMS-151-10100 single layer Lidars used as a safety curtain, two Velodyne VLP-16 16-layer Lidars to provide a 3D view in front of and behind the shuttle, and two SICK LD-MRS 4 layer Lidars for long-range localization. The safety Lidars are used to detect nearby obstacles that the vehicle could collide with and activate the emergency stop using low-level hardware connections. The two Velodyne Lidars are mounted at the front and rear of the shuttle to create a large point cloud, which we use for obstacle detection and 3D tracking. The two long-range Lidars on top of the shuttle provide localization data in forward and backward direction.

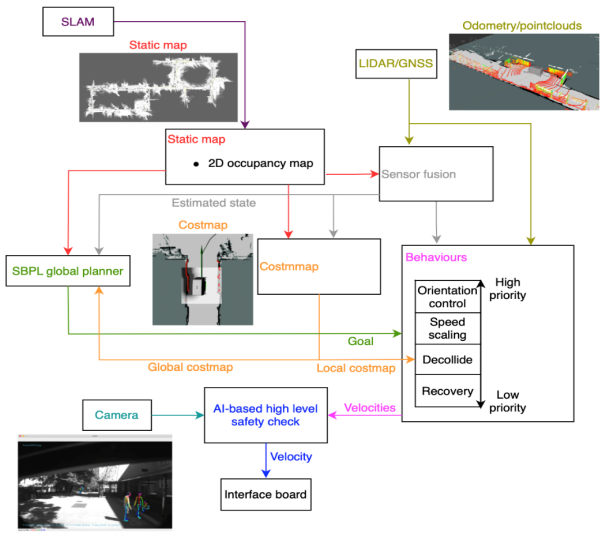
2.4 Cameras

The bus comes fitted with two Flir Point Grey cameras which produce good quality grayscale images. At this stage of development, grayscale images are sufficient for the Neural Network learning process and obstacle tracking features. Each camera is configurable to use different size images and publishing rates to avoid flooding networks and reducing processing times on images.

3. Vehicle software structure

The development of the software stack for an autonomous vehicle can be achieved in several ways. For most commercial projects, software teams would spend months building a full stack solution on socket communication, device drivers, and basic movement algorithms – before any of the real development work of the project can begin. This approach leaves development teams with high startup costs and limits the flexibility to change systems and platforms depending on the initial development. An alternative approach used mostly in research but growing in industrial use, is the Robot Operating System (ROS) (Macenski, et al., 2022). ROS gives development teams the base level stack needed for communication and resource handling. So, feature development can begin within days not months. In addition, ROS comes with many software packages and common drivers which are readily available, reducing the development time needed and providing flexibility when changing platforms or devices.

Figure 2: nUWAy software structure using ROS2



The nUWAy shuttle bus software stack (Fig. 2) is built using the ROS2-humble distribution as it is the current long-term release of the ROS2 software and provides several additional features and improvements. The software follows a typical navigation stack for mobile robots and varies based on driving techniques, but the structure remains similar. Fig. 2 shows the network stack for a SLAM (Simultaneous Localization and Mapping) (Macenski & Jambrecic, 2021) implementation of driving which is discussed in later sections. The goal of the stack is to break the code into meaningful parts to make it easier to control the data flow and debug issues.

4. Waypoint navigation

The Regulated Pure Pursuit (RPP) algorithm (Macenski & Singh, 2023) was used to achieve exact path following. We have implemented the Pure Pursuit algorithm with added velocity regulation to improve driving efficacy and safety. This includes regulating the linear velocity on high curvature paths and on proximity with an obstacle.

For path planning, we used the Smac-Hybrid planner (Macenski, 2023) which is an implementation of a Hybrid A* path planning algorithm. The Smac planner is a time-efficient implementation of the A* algorithm that considers only paths that the bus can feasibly achieve.

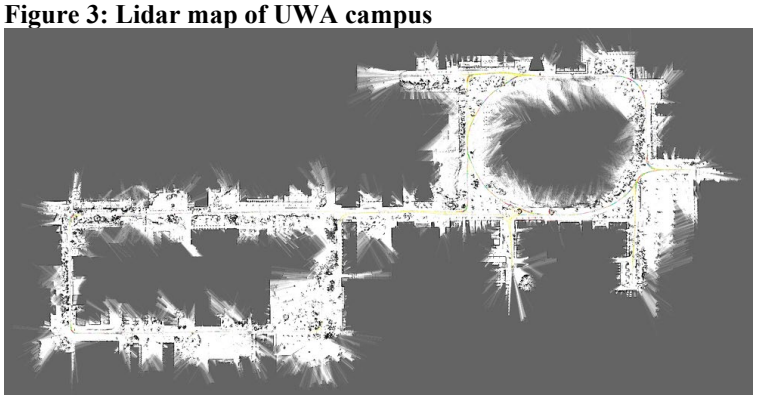
We are currently using two test locations, one at the UWA Crawley campus and one on Cinnabar Drive, Eglinton, which is the new Amberton Beach residential development. At the Crawley test site, we used an AUSCORS (Australian Government - Geoscience Australia, NA) base station to provide RTK corrections (Radio Technical Commision for Maritime Services, 2004). For the Amberton Beach trial we set up our own base station using a surveying grade GPS antenna that collects data in a 24-hour period to get a highly accurate position (Government of Canada, 2021). The corrections are then forwarded to the shuttle buses using RTK2go (SubCarrier Systems Corp., 2021). We found that without corrections, we could achieve sub-meter accuracy of around 0.7m in our testing, but this value ranged greatly up to 2m. With the RTCM corrections, we achieved accuracies of 2cm when GNSS signal was readily available from clear skies and no obstructions.

5. Lidar-based navigation

The basic task of a self-driving vehicle is to navigate itself from an initial place to the destination autonomously. To achieve this goal, a complete autonomy system is required to handle sub-tasks in autonomous driving such as map building, localization, global and local path planning, etc. (Badue, et al., 2021). The nUWAy shuttle bus can perform these subtasks based on Lidar data. The campus map created by our offline method is used to construct a global cost map (Lu, et al., 2014). This cost map then gives the global path planner a sense of where it is easy to drive. The bus locates itself in the campus map by fusing our Lidar odometry and GNSS data. The Lidar data is used to build the local cost map which records the dynamic obstacles in the surroundings. This cost map and localization result goes into the local planner module to generate the primitive driving commands. The command stream would be finalized based on a high-level AI safety check node.

An open-source 2D SLAM technology Google Cartographer (Konolige, et al., 2010) (Hess, et al., 2016) has been integrated into our nUWAy shuttle bus, which is using Lidar measurements to build an offline campus map. The campus 2D map from Reid Library to the business school is shown in Fig. 3.

Figure 3: Lidar map of UWA campus

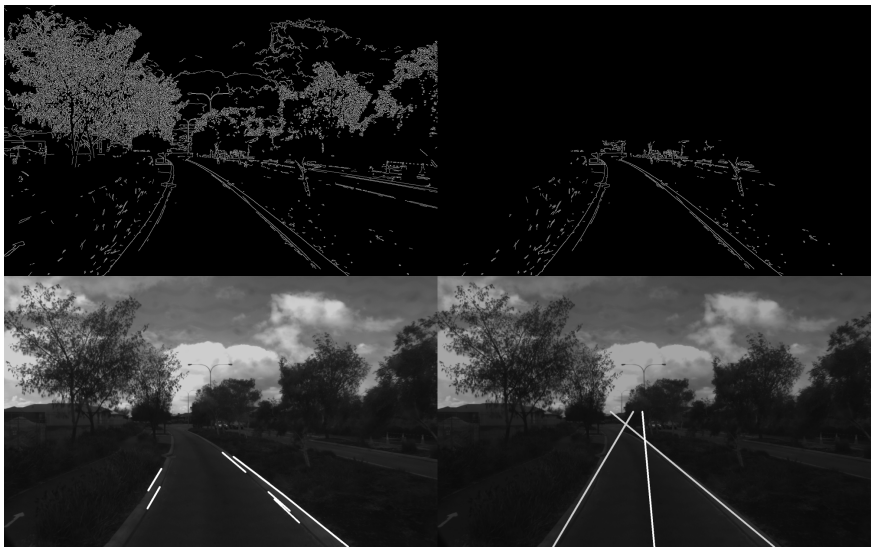


6. Vision-based navigation

Vision-based navigation can be implemented using traditional Engineering methods or by using more recent deep learning methods. The traditional Engineering method is based on programmed computer vision operations, e.g., software packages like OpenCV, to detect lane markings and curbs and then plan a driving path according to that.

Our current algorithm is applying Canny edge detection and a subsequent Hough line detection algorithm to identify the lane's left and right boundaries, to determine the shuttle's steering and throttle commands (Fig. 4). However, this approach only works on straight and curved road segments and cannot yet handle intersections or roundabouts. These need to be handled by the safety operator for the time being.

Fig. 4: OpenCV workflow showing: top left - edges, top right - region of interest, bottom left - superimposed lane markings, bottom right - smoothed lane borders and heading

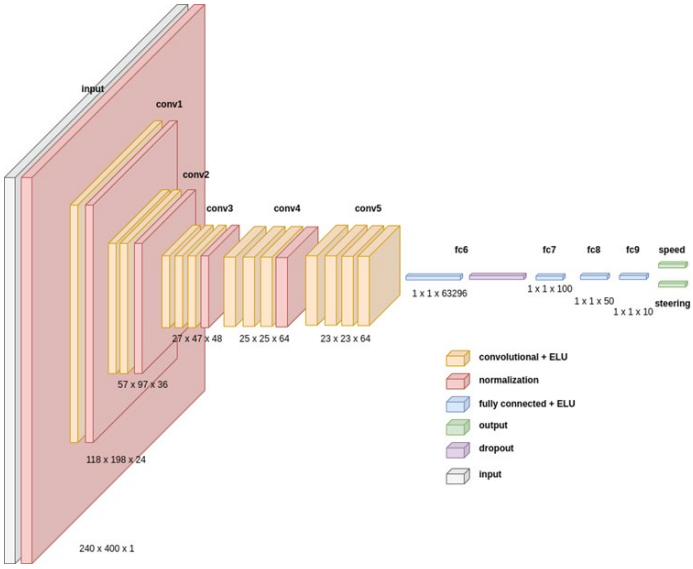


7. AI-based navigation

A deep-learning method is based on an artificial neural network approach, for which we use the TensorFlow software package to train a neural network with image input data. We used a modified version of PilotNet (Bojarski, et al., 2017) (Fig. 6) that takes the front camera image as input and outputs the corresponding steering angle and throttle control commands (Fig. 5). This method can handle intersections with traffic lights and is implemented in real shuttle buses.

Fig. 5: NN workflow showing steering angle and throttle control commands in the test dataset. (gray line on the left represents actual commands, white line on the right represents neural network commands, line length is controlled by throttle, line angle is controlled by steering)

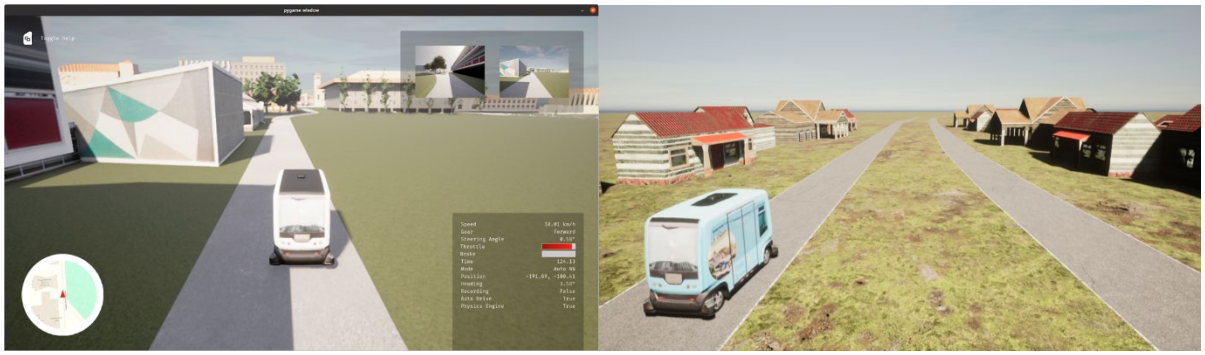
Fig. 6: End-to-end NN architecture



8. Shuttle simulation system

In addition to the real-world vehicle, the REV team has set up a simulation environment for an autonomous shuttle bus using Carla (Dosovitskiy, et al., 2017). Initially developed as a hardware-in-the-loop (HIL) system (Brogle, et al., 2019), it was later found that the HIL component did not enhance the system's performance or accuracy, so it was converted into a pure software simulation. In the simulation environment, a digital version of the entire UWA university campus and Amberton beach was developed, along with a model of the autonomous shuttle bus (Fig. 7). The simulator has two manual control inputs: A steering wheel with accelerator and brake pedals and a keyboard input. With this system, users can easily set up and conduct driving experiments.

Figure 7. Shuttle bus simulations for on campus and at Amberton Beach



To operate the simulator, we incorporated five distinct modes:

1. Manual mode, where the vehicle is controlled manually with joystick or keyboard.
2. Lidar autonomous mode, where the Lidar-based navigation stack drives the vehicle.
3. Computer vision autonomous mode, which uses OpenCV-based vision algorithms to operate the shuttle.
4. Neural network autonomous mode, which employs a deep learning neural network model for control.
5. Mirror mode, which utilizes the GPS coordinates of the real vehicle in the simulation.

9. Amberton Beach data collection

The road trials require extensive data capture systems, in particular systems that can capture and document failures/incidents. A recording system has been implemented for the shuttle bus in the ROS2 operating system (Macenski, et al., 2022). The system comprises two systems:

1. Regular path recorder
   This system records the shuttle's exact position using RTK-GPS (Cannon, et al., 2001) with timestamps. With this, we are able to document and analyze the shuttle's paths with centimeter-accuracy. An example is given in Fig. 8.

2. Incident recorder

   The incident recorder will automatically be activated whenever the vehicle's safety system triggers an emergency stop (detected through the low-level CAN bus system), the high-level navigation software shuts down, or when a user pushes one of the emergency buttons. Each of these events is classified as a "disengagement" of the autonomous driving system, which requires reporting and investigation. The system records all messages, including camera, Lidar, GPS, CAN bus, etc., for a sequence of time of about 10 seconds. The incident data can be post-processed using third-party software like Foxglove (Foxglove, 2023). For reporting purposes, the vehicle position and orientation as well as its safety-Lidar data will be drawn into a map similar to the web-based vehicle location service (Fig. 9).

Figure 8: Regular recording figure

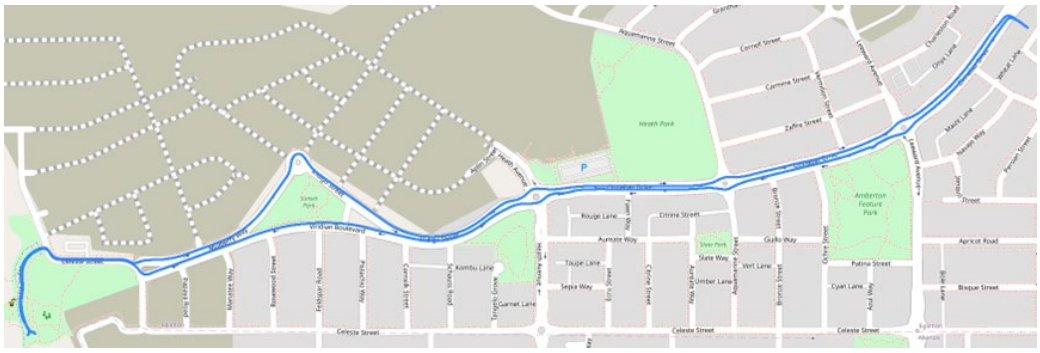


Figure 9: Web-based vehicle location service

The regular recorder records global position and kinematic parameters all the time, and records GPS position and CAN data every 10 meters to monitor the autonomous driving system. The recorded data is used to locate problem areas in the current driving implementation as well as to improve path planning and detection, train control models, and path accuracy. Additionally, the data can be used to evaluate the driving status and road conditions. Overall, this recording system provides a valuable tool for analyzing and improving autonomous driving systems.

10. Results

The autonomous shuttle bus has been tested with a variety of control methods. Driving data provides valuable insights into where current implementations are failing and how they might be improved. Road training in Amberton Beach began earlier this year with data collection from manual driving. This data provided the basis for extending a campus-based autonomous vehicle to a road-based solution. The following sections summaries the data collected so far, with a comparison of methods as well as the current limitations of the technology.

10.1 Simulated driving

The simulation system provides a safe space for testing software before it is deployed in the real world, this is particularly important for AI solutions where the control output is not always known. The simulated shuttle bus has currently been set up to use either a neural network or computer vision for testing in various campus-based situations including turning corners and decision making at intersections. The results of the simulator are below.

Table 1: Simulation results

	Neural Network	Computer Vision
% Time Autonomous Mode (AM)	98.73%	64.96%
% Time Manual mode (MM)	1.27%	35.04%
Number of interventions	1	17
Number of times out of bounds	5	22

Limited testing has been done with GPS and SLAM based solutions as they require additional sensor setup as well as random error additions. Looking at the results, it is clear that neural networks provide a sound solution for autonomous driving.

10.2 UWA campus driving

Currently the shuttle bus at the UWA campus has three different methods of navigation, which are being tested and compared to make future improvements. Table 2 comparing the results from deployment using pure GPS. GPS+RTK, SLAM and neural networks. Multiple trips were taken following the main path from the Ocean Marine Centre to Business School and back as this is one of the major routes planned for future services.

Figure 10: Map showing major stops on the UWA campus

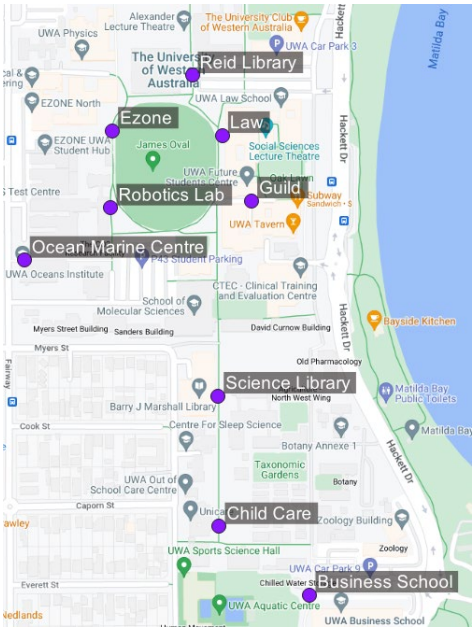
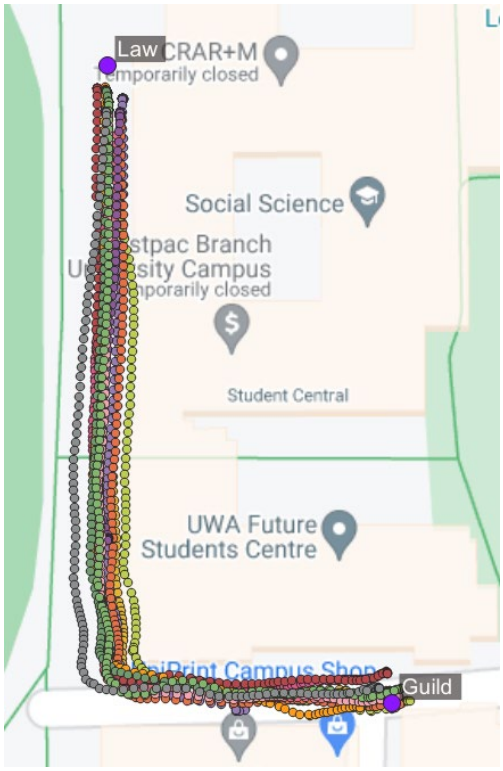


Table 2: Real driving results

	GPS	GPS+RTK	SLAM	NN
% Time AM	41.78%	56.19%	83.33%	73.32%
% Time MM	58.22%	43.81%	16.67%	26.68%
Number of interventions	5	9.5	30	18

The time spent in autonomous and manual mode is given as a percentage as each journey's average time can vary depending on the type of intervention and wait time. The neural network and SLAM implementations currently have the most stable runs, typically running in autonomous mode for much of the journey. The neural network model is more reliable, however with a smaller number of interventions and is currently limited by its training around obstacle avoidance. The SLAM implementation has a high autonomous rate but with the current implementation has a higher failure rate. This is due to the SLAM implementation delocalizing from the map and moving towards obstacles or locations where the vehicle is not allowed to run. Fig. 11 shows a shorter journey between law and guild where the shuttle bus has performed well yet due to localization issue it will deviate from the planned path. At its largest the bus deviated approximately 1.5m from its planned path, which is the gray path shown, however the system was able to recover after the corner due to better map localization at this point.

Figure 11: Map showing path deviation using SLAM



Comparatively GPS and GPS+RTK implementation has far fewer interventions, but this could also be attributed to the longer manual control where the operator is waiting for a clear path. The GPS model also struggles to stay on the planned path due to the error for the GPS coordinates. GPS+RTK shows an improved performance when compared to pure GPS due to the much higher coordinate accuracy allowing it to follow the path more rigidly. There are still, however, several locations along the path where the vehicle struggles to drive using GPS+RTK, such as UWA's Science Library, where buildings cause signal interference and the vehicle starts

to veer away from the path, requiring intervention. The SLAM implementation has proven to overcome these issues when it can properly localize, which requires larger stationary structures nearby.

10.3 Pedestrian detection on UWA campus

Further to operational performance, we have implemented a pedestrian detection system whose primary task is to detect people for collision avoidance. However further work is being performed to implement a gesture detection system to improve human-machine interaction. The pedestrian detection system has been implemented in YOLOv7 which can detect users in real time scenarios. Figure 12 shows a successful detection of a pedestrian walking past the vehicle.

Table 3: Pedestrian detection accuracy

Scenario	IoU (Intersection over Union) (%)
Sparse	67.7
Dense	57.1
Total	59.7

Figure 12: Successful detection of walking pedestrian

Due to the light exposure and natural shrubs this detection process still needs improving. The following images show two current failure modes of the detection system. The image on the left shows a tree being misclassified as a human and the image on the right shows undetected pedestrians in a small crowd. Further training of the detection model is needed to make this system reliable.

Figure 13: Tree detected as a pedestrian (left); undetected pedestrians (right)

10.4 Amberton beach driving

During the initial manual driving stage, data has been recorded for path accuracy, image data for neural networks and the development of safety systems. At this stage only the neural network mode has been tested with the following results.

Table 4: Neural network driving accuracy

	Neural Network
Percentage of Time Autonomous Mode	43.96%
Percentage of Time Manual mode	56.04%
Number of interventions	11

The model developed for Amberton Beach is still in its infancy and requires more data for improvement. The system has been designed to drive up to intersections where on-board safety operators will determine when it is safe to proceed through. It is a development goal to automate this, but more sensor data is required to do this safely. A GPS+RTK drive system has also been developed and will be combined with the neural network.

11. Conclusions

We have presented the implementation of an autonomous shuttle bus showing the current limitations of the implemented systems. While we work to continuously improve these systems, we have begun conducting passenger drives on the UWA Crawley campus and will work with the UWA Business School on how we can improve the uptake and usage of these systems. The system has been built using open-source software packages from ROS2, combined with software written by students. On the university campus with its dense building structure, we have found that Lidar-based navigation and vision-based navigation using neural networks to provide the best results. For the shuttle operation on public roads at Amberton Beach, we are implementing a combination of RTK-enhanced GNSS in combination with a vision-based deep learning approach. These new algorithms are first being tested in our autonomous vehicle simulation system before they get deployed on the actual shuttle bus. Another goal is to develop an on-demand system that will take users on the initial or final leg of their journey.

Acknowledgements

We would like to thank our REV sponsors Stockland, Allkem, Tesla Slack, CD Dodd, Dyflex, Tesla Roadster Inc, Tesla Owners Club of WA, Tesla Owners Club of Australia, SBG Systems, and Altronics, as well as our university sponsors UWA School of Engineering, Business School, School of Physics, Mathematics and Computing, and the UWA Innovation Quarter.

References

Anderson, B. & Moore, J., 1979. *Optimal Filtering.* New Jersey: MIT Press.

Australian Government - Geoscience Australia, NA. *AUSCORS NTRIP Broadcaster.* [Online] Available at: https://gnss.ga.gov.au/stream [Accessed April 2023].

Badue, C. et al., 2021. Self-driving cars: A survey. *Expert Systems with Applications,* Volume 165, p. 113816.

Bojarski, M. et al., 2017. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 25 April.

Bräunl, T. et al., 2022. *Building an Autonomous Drive System for an Electric Shuttle Bus.* Melbourne, SAE Australasia, p. 7.

Brogle, C., Zhang, C., Lim, K. & Bräunl, T., 2019. Hardware-in-the-Loop Autonomous Driving. *IEEE Transactions on Intelligent Vehicles,* 4(3), pp. 375-384.

Cannon, M. E. et al., 2001. *GPS RTK Positioning using a regional reference network: theory and results.* Seville, Global Navigation Satellite Systems.

Dosovitskiy, A. et al., 2017. *CARLA: An Open Urban Driving Simulator.* Mountain View, PMLR, pp. 1-16.

Drage, T. et al., 2021. *Integrated Modular Safety System Design for Intelligent Autonomous Vehicles.* Nagoya, IEEE, p. 8.

Foote, T. & Purvis, M., 2010. *REP - 103.* s.l.:ROS.

Foxglove, 2023. *foxglove.* [Online] Available at: https://github.com/foxglove [Accessed April 2023].

Government of Canada, 2021. *About the Canadian Spatial Reference System.* [Online] Available at: https://natural-resources.canada.ca/maps-tools-and-publications/tools/geodetic-reference-systems/canadian-spatial-reference-system-csrs/9052 [Accessed April 2023].

Hess, W., Kohler, D., Rapp, H. & Andor, D., 2016. *Real-time loop closure in 2D LIDAR SLAM.* Stockholm, IEEE.

Konolige, K. et al., 2010. *Efficient sparse pose adjustment for 2D mapping.* Taipei, IEEE.

Lim, K. et al., 2018. *A Modular Software Framework for Autonomous Vehicles.* Chang Shu, IEEE, pp. 1780-1785.

Lu, D., Hershberger, D. & Smart, W., 2014. *Layered costmaps for context-sensitive navigation.* Chicago, IEEE, pp. 709-715.

Macenski, S., 2023. *Smac Planner.* [Online] Available at: https://github.com/ros-planning/navigation2/tree/main/nav2_smac_planner [Accessed April 2023].

Macenski, S. et al., 2022. Robot Operating System 2: Design, architecture, and uses in the wild.. *Science Robotics,* 7(66), p. eabm6074.

Macenski, S. & Jambrecic, I., 2021. SLAM Toolbox: SLAM for the dynamic world. *Journal of Open Source Software,* 6(61), p. 2783.

Macenski, S., Martín, F., White, R. & Clavero, J., 2020. *The Marathon 2: A Navigation System.* Las Vegas, IEEE.

Macenski, S. & Singh, S., 2023. *Regulated pure pursuit controller.* [Online] Available at: https://github.com/ros-planning/navigation2/tree/main/nav2_regulated_pure_pursuit_controller [Accessed April 2023].

Meeussen, W., 2010. *REP - 105.* s.l.:ROS.

Radio Technical Commision for Maritime Services, 2004. *The Radio Technical Commission for Maritime Services (RTCM) Publishes New Standard for Networked Transfer of RTCM via Internet Protocol (Ntrip).* [Online] Available at: https://igs.bkg.bund.de/root_ftp/NTRIP/documentation/NtripPressRelease.pdf [Accessed April 2023].

SubCarrier Systems Corp., 2021. *RTK2go.* [Online] Available at: http://rtk2go.com/ [Accessed April 2023].