|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **IET Integrated Development Scheme Graduate Logbook Entry** | | | | | | | | | |
| Reference No: | | | REEL 006 | | | | | | | | | |
| Graduate Name: | | | Jack Haworth | | | | | | | | | |
| Department/section: | | | Robotics in Extreme Environments Lab | | | | | | | | | |
| Report Title: | | | Integrating GPS System into CARMA | | | | | | | | | |
| Dates/Duration: | | | Jul-Nov 2020 | | | | | | | | | |
| **Competence Development** | | | | | | | | | | | | |
| This report covers development of the following UK-SPEC competencies | | | | | | | | | | | | |
|  |  | | | |  | | | |  | | |  |
|  |  | | | |  | | | |  | | |  |
|  |  | | | |  | | | |  | | |  |
|  |  | | | |  | | | |  | | |  |
|  |  | | | |  | | | |  | | |  |
| Includes evidence of Further Learning ? | | | | | |  | Learning hours claimed? | | | | | hrs |
| See section 4 of the IET graduate scheme manual for details on further learning – **Double check this?** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Declaration**  I certify that I have read the attached report and confirm that it is, to the best of my knowledge and true and accurate statement. | | | | | | | | | | | | |
| Signed (Graduate): | | | | | | | | | | Date: | | |
|  | | | | | | | | | | | | |
| **Submission Record** | | | | | | | | | | | | |
| Submitted to: | | | | Harriet Peel | | | | | | | | |
| Position/Job Title: | | | | REEL Roboticist/Supervisor | | | | | | | | |
| Signed:A picture containing shape  Description automatically generated | | | | | | | | | | Date: 21/10/20 | | |
|  | | | | | | | | | | | | |
| **Mentor Assessment** | | | | | | | | | | | | |
| Mentor: | | | | Stephen Parkinson | | | | | | | | |
| Comments | | | | | | | | | | | | |
| Signed: | | | | | | | | | | Date: | | |
| Office use | | | | | | | |  | | | | |
| Further Learning Assessment Panel Members | | | | | | | |  | | | | |
| M-Level credit equivalence | | | | | | | | Credits (learning hours/10) | | | | |
| FL Panel Date | |  | | | | | | Panel Initials | | |  | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Summary -** A short summary of the activity, including key experiences, lessons learned and how you have deployed the experience gained | | | | | | | | | |
| This activity details the understanding obtained of the Outdoor CARMA (Continuous Automated Radiation Monitoring Assistance) robotic platform in terms of software and hardware in order to integrate GPS (Global Positioning System) for autonomous navigation of Calder Hall on Sellafield.  GPS technology is beneficial for accurate outdoor navigation, but since CARMA is used to detect potential areas of radioactive contamination, the GPS signal cannot be transmitted over any networks that could, under unlikely circumstances, be compromised by external parties. Security has been taken into consideration at every stage during the design of the CARMA platform, but security considerations are especially important to consider for GPS and radiometric data.  For security reasons, transmitting GPS signals is not allowed on this platform because if compromised this could potentially allow hackers to obtain the exact location of radioactive materials through the interpretation of the GPS co-ordinates in line with the radiation map. Therefore, as part of this task I had to write some ROS (Robot Operating System) code in Python so that the positional information transmitted, is an odometry type message, not GPS. This is so that the location is only an x, y, z co-ordinate in meters relative to the base station, not an absolute position.  The hardware was assembled and integrated to allow the functionality to be tested prior to the upcoming demonstration. | | | | | | | | | |
| **Relevant Competences-** Reference to the applicable engineering council competence(s) that the activity may have developed or provided an opportunity to demonstrate | | | | | | | | | |
| A1 | X | B1 | X | C1 |  | D1 |  | E1 |  |
| A2 | X | B2 | X | C2 |  | D2 |  | E2 |  |
|  |  | B3 | X | C3 |  | D3 |  | E3 |  |
|  |  |  |  | C4 |  |  |  | E4 |  |
|  |  |  |  |  |  |  |  | E5 |  |
| **Introduction-** An introduction providing the context of the task | | | | | | | | | |
| CARMA is an autonomous robot based on the Clearpath Jackal platform. This outdoor version is set to be deployed to perform radiometric surveys around Calder Hall, Sellafield, to declassify the surrounding compounds ~~out the area for radiation~~. To allow for a better understanding of the outdoor CARMA, a diagram of the along with a link to a video showcasing a previous iteration of the robot can be found in Appendix A1. The CARMA platform is equipped with a Thermo Scientific DP8 Alpha-Beta probe, which is integrated into the mapping software to detect and report areas of contamination.  CARMA also uses a LiDAR (Light Detection And Ranging) sensor, cameras and wheel encoders for autonomous navigation. LiDAR is a very popular method for measuring distances (ranging) by illuminating the target with laser light and measuring the reflection with a sensor. Differences in laser return times and wavelengths can then be used to make digital 2-D or 3-D representations of the target. However, LiDAR is not always reliable in an unknown outdoor environment as it relies on reference points to be able to know where it is, therefore if the there are no clear reference points, or they are too far away, it may prove difficult to create the maps required to localise and navigate, varying levels of sunlight can also interfere with LiDAR mapping. Wheel odometry can also be unreliable, especially on outdoor terrain where it may be subject to wheel slips etc. therefore a combination of all three systems is used to provide a sub centimetre accurate localisation system for autonomous navigation.  Usually GPS is only accurate to around 5-10m. However, this system uses a GPS Base Station which will be located nearby, this communicates with the GPS unit on CARMA and uses Real Time Kinematics (RTK) to reduce the error by comparing the GPS signals of the Rover and the Base station with another data source, such as radio. By having two GPS units in situ this allows the ionospheric error to be cancelled out as both the Base Station and Rover will be subject to the same error. This is the largest error associated with GPS systems and as such allows the Rover (CARMA) to calculate its position with respect to the base station, eliminating this error, allowing for a much more accurate position to be obtained.  A glossary can be found in appendix A2 to aid with understanding. | | | | | | | | | |
| **Objectives-** Objectives of activity, assignment or project | | | | | | | | | |
| 1. Understand how CARMA works. Complete ‘Mastering with ROS: Jackal’ course. 2. Understand how the GPS System works. 3. Configure the GPS System for CARMA. 4. Integrate software and hardware. 5. Test functionality of GPS system on CARMA. 6. Write code to convert GPS Signals to Odometry 7. Test functionality of modified ROS code and Odometry output when running CARMA. | | | | | | | | | |
| **Description-** Description/detail of activity, assignment or project | | | | | | | | | |
| 1. The first thing I did, whilst still not working in the lab, was complete ‘Mastering with ROS: Jackal’ course on The Construct (Same as ROS Basics). This is the platform on which CARMA is built so this was a good foundation to be able to understand how it operates. The learning outcomes for this course can be seen in Appendix A1. 2. When undertaking this course, I particularly focussed on the section on how to convert normal GPS (i.e., not using RTK) data to Odometry in order to fuse it with wheel odometry and IMU (Inertial Measurement Unit) using a Kalman Filter in order to increase accuracy of the positioning system. 3. I then read all literature on the COTS (Commercial Off The Shelf) GPS System (Piksi Multi) to be integrated into CARMA. Piksi Multi is a multi-band, multi-constellation RTK (Real Time Kinematic) GNSS (Global Navigation Satellite System) receiver board that provides centimetre-level accurate positioning. 4. I then followed tutorials to configure the system for both CARMA and the Base Station. 5. I then integrated the hardware into CARMA, to do this I had to redesign the GPS mounts on the battery box to allow me to fix the GPS Receiver to the top of the box. I also repositioned the 12v power rail to reduce cable run lengths and made up and installed new power cables for the GPS system. 6. I then read the ROS Integration documentation in order to install the required ROS packages from GitHub repositories. 7. I digested the various topics and messages that the Piksi Multi ROS System uses to understand how I could modify the package to output an Odometry type message. 8. I found by using the command ‘rostopic echo /piksi/navsatfix\_rtk\_fix I could get the longitude and latitude co-ordinates from base\_link (this is the middle of the robot). 9. Using the learning from ‘Mastering with ROS: Jackal’ course I was able to use the geonav transform to convert the longitude and latitude to x, y co-ordinates as an Odometry type message. 10. However, from further research into the Piksi Multi ROS code and trialling in simulation mode I found a more effective way of creating an Odometry type message which allowed the GPS topic to be disabled completely. 11. This makes the system much more secure as the GPS data is not being published, this removes the potential of the GPS being reconverted to give absolute position. 12. I found that I could get the PoseWithCovariance directly from the Piksi Multi, which gives me x, y, z co-ordinates in meters, respective to the base station. The problem with this however is that it is a geometry type message, not odometry. 13. Odometry is the use of data from motion sensors to estimate change in position over time. 14. In order to get Odometry I needed to add Velocity to PoseWithCovariance. Unfortunately, velocity is not published in simulation mode so I had to build the system into CARMA and test it outside to ensure Velocity data would be published. 15. I created a new function in the main script where I integrated the two messages (Velocity NED (North East Down) and PoseWithCovariance and published them as a new topic made up of a nav\_msgs/Odometry type message incorporating velocity and position. This can then be subscribed to, producing a centimetre accurate location of the robot in meters, respective to the base station. 16. Appendix A3 shows a breakdown of the code added to the piksi multi source file. Each section has been annotated to explain its function. 17. I was then able to fuse the output of this new topic into the Kalman Filter with the wheel odometry, LiDAR unit and IMU to improve the positional accuracy and velocity control of CARMA. This can be seen in Appendix A6. 18. It is important to also utilise wheel odometry in order to keep the motion smooth as LiDAR and GPS take longer to respond which is likely to result in the robot displaying a jerking motion due to the delay. 19. I also had to modify the URDF (Unified Robot Description Format), as seen in appendix A5, to incorporate the GPS sensor into the physical robot model. The URDF is an XML file which represents the physical robot model and its sensors etc. The parent link is set as top\_box\_link, as this is where the GPS Receiver is mounted, the offset translates the co-ordinates from the base of the top\_box to the base of the GPS Receiver. 20. The base\_ned frame below this is a child link of the GPS Receiver. This incorporates the Odometry data published by the new function and topic I created. This URDF can now be added into the main CARMA URDF and will allow this to be translated back through the links and frames to be representational of the robot base\_link frame. 21. CARMA was then retested to ensure functionality. 22. The target date for this was for a live demonstration on 9th November 2020. However, due to COVID-19 the demonstration was cancelled. It was decided that a video would be created instead, allowing CARMA to be showcased to a wider audience and minimising risk. I will be contributing to this in the coming weeks. | | | | | | | | | |
| **Conclusions-** Conclusions on how the activity develops engineering competence. | | | | | | | | | |
| A1 – Here I identified the limits of my own personal knowledge and skills and strove to bridge this gap by completing the Mastering with ROS: Jackal course and researching and learning about GPS Systems and ROS Integration.  A2 – Used a sound evidence-based approach to problem solving from learning ROS and understanding how the systems operate to allow me to problem solve and contribute to continuous improvement of the CARMA platform.  B1 – Identified and reviewed different methods for outputting secure positional data from the GPS system and developed the design requirement. Identified the risk of converting GPS data as oppose to eliminating GPS data as it could potentially be converted back. Carried out necessary tests.  B2 – Designed, developed and tested my own code to create an Engineering Solution to a robotics task.  B3 – Implemented and tested my own design solutions. A level of debugging and testing of the code was also required to achieve this. Other parameters also required adjusting, such as amending the baud rate in the launch file to match the Piksi Module. | | | | | | | | | |
| **Appendices** | | | | | | | | | |
| A1 - image from <https://uomrobotics.com/robots/carma-2.html>    Video can be seen here: <https://www.youtube.com/watch?v=2W7EHUvoBA4> I am also currently working on a new promotional video to showcase outdoor CARMA.  Appendix - A2  **Glossary**  CARMA – Continuous Automated Radiation Monitoring Assistance  COTS – Commercial Off The Shelf  GNSS – Global Navigation Satellite System  GPS – Global Positioning System  IMU – Inertial Measurement Unit  LiDAR – Light Detection and Ranging  NED – North East Down  Odometry – the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location.  PoseWithCovariance – Position (x,y,z) with Covariance  RGB – Red, Green and Blue  ROS – Robot Operating System  RTK – Real Time Kinematics  URDF – Unified Robot Description Format  XML – Extensible Markup Language  Appendix - A3   1. How to set up the navigation stack to make it navigate in an indoor environment, generating maps on its own. 2. How to create a program to navigate in outdoor environments through GPS data. 3. How to detect persons with the laser sensor. 4. How to detect persons with its RGB stereo camera. 5. How to generate waypoints and make jackal patrol. 6. Use the StereoCam to generate PointCloud Data 7. Create a reactive program based on all previously mentioned and create a patrolling program that reacts to person detection. | | | | | | | | | |
| A4 - Annotated Code   1. Imported Odometry from the nav\_msgs.msg library      1. This creates the velocity variables as objects and sets their initial state to 0. They are later referred to as ‘Global’ Variables, although these are not actually global as they are only used within this class.      1. Created a publisher for the Odometry topic      1. Created a function to set the velocity variables using the velocity NED topic data.      1. Created a function to combine velocity and position and publish the data as an Odometry type message.        1. Created callbacks to these functions in order to activate them. | | | | | | | | | |
| Appendix A5  GPS URDF Addition | | | | | | | | | |
| Appendix A6  Kalman Filter – Fusing Odometry Sources | | | | | | | | | |