A Very-Low-Cost GNSS Precipitable Water Vapour Sensor

1 Project Progress Update

The goal of the project is to develop a very-low cost system for the measurement of precipitable water vapour from GNSS data utilizing inexpensive, off-the-shelf, commercially available components. The guiding principles for the design of the system were first-and-foremost that it be low cost, but also that it could be deployed to remote areas which may lack infrastructure such as internet connections. So, wherever possible, the system was designed to be able to function autonomously after some initial configuration.

The project was conceived of as having five milestones, with the goal being to complete the first three: hardware assembly, software development, and validation of the system, during the Spring 2018 semester. Hardware assembly was successfully completed and significant progress was made on software development, but it was not completed, delaying validation of the system.

While it has been delayed, my intention is to continue work on the project through the summer (albeit at a lower intensity) as a personal project, as the current challenges remaining in the software development are both modular (easy to work on in pieces) and interesting. To that end, below I will offer an assessment of the work completed over the semester, as well as outline some of the new potential directions and opportunities which have arisen during the semester.

1.1 Milestones

1.1.1 Hardware: GNSS Sensor Assembly

Proposed Milestone: Hardware will need to be ordered, assembled, and troubleshot. The Raspberry Pi single-board computer will be used to collect, process, and store data from the GPS receiver. The Raspberry Pi can be ordered as a kit containing a power supply, memory cards, heat sinks, and a case. The NEO-M8T GPS receiver is an independent breakout board which will will be connected to the Raspberry Pi via USB. If the Raspberry Pi case has sufficient space, the NEO-M8T will be secured and mounted to it, if not, a custom case or structural support will be fabricated from plastic or wood.

Deliverable at this milestone will be a functioning computer controlled GPS receiver with structural support. Goal: mid-March 2018.

Evaluation: Hardware assembly was completed successfully but evolved slightly from the initial conception. The Raspberry Pi was ordered off of Amazon, and arrived promptly. It's setup was straight forward, requiring the adhesion of some heat sinks and some initial operating system configuration. The NEO-M8T GNSS receiver was ordered from CSG Shop, which is located in Latvia, so shipping took slightly longer than expected. The receiver required an initial configuration using software provided by the manufacturer in order to enable the output of the raw GNSS measurements. While the bare sensor board was capable of detecting a small number of satellites (up to 2 in a challenging urban environment), which was suitable for testing and development, I decided that an antenna was required to achieve suitable performance. An active antenna that is compatible with the NEO-M8T was ordered from CSG Shop, and after installation, the sensor was able to detect up to 20 satellites in the exact same environment. Additionally, since some of the models used in the water vapour estimation process require local temperature and pressure measurements, an inexpensive ($8) thermometer/barometer unit was purchased. During development, these temperature and pressure values can be accessed from the internet, but if this system is to be deployed remotely, it may be necessary to take these measurements on it's own.

My main takeaways from the hardware component of the project are:

1. The Raspberry Pi is a suitable platform upon which to build this sensor system.
2. The ublox NEO-M8T GNSS receiver and ublox active antenna can provide the necessary raw data measurements needed to estimate atmospheric water vapour.
3. Initial assembly and configuration of the hardware components is straight forward and should not pose a hurdle to anyone interested in building their own version of this system.

1.1.2 Software Development

Proposed Milestone: Development of a software package that can collect, process, and store GNSS data be the main portion of the project. Development will be done in a combination of C and C++. The data acquisition (DAQ) component of the package will be code which controls and receives raw GNSS measurements from the GPS receiver. There are open-source libraries such as RTKLib which can be used to parse messages from the receiver. Relevant measurement data will be harvested from the parsed messages and formatted for processing into PWV measurements. The DAQ will be written in C++, utilizing C to communicate with the GPS receiver. A data processing program will be written in C++ to estimate PWV values from the raw measurement data according the the methods outlined by, for example, Yuan *et al.*, and output this data into a human-readable format [1]. Control and automation of the DAQ and data processing components will be done with bash scripts in the Raspberry Pi's unix environment.

Deliverable at this milestone will be a functioning GNSS sensor with the capability to take PWV measurements. Goal: end of April 2018.

Evaluation: While substantial progress was made on software development, this milestone was not completed as a whole. The RTKLib libraries proved difficult to use, instead the *ublox* library (https://github.com/arobenko/ublox) was used to interface with the GNSS receiver (the manufacturer of the NEO-M8T does not provide easily modifiable code with which to interact with their receivers). Since the ublox library is maintained by a single developer as a personal interest project, it is not as thoroughly documented as I would like, but after some amount of time digging through the code and the NEO-M8T manufacturer's documentation, I was able to implement the communications protocols into my own code. Additionally, I used the *boost* libraries (https://www.boost.org/), a set of portable (behave the same across different platforms) free C++ libraries to handle serial port communication (the physical link between the computer and the GNSS receiver). My code (and installation instructions) are on the MIT Github (https://github.mit.edu/tdmacd/ordeal) to facilitate development, however the MIT Github is not publicly accessible without MIT certificates, so it will need to be migrated to the public Github once it is ready.

The capabilities of the software can be roughly broken into to components, the ability to receive and record raw GNSS measurement data from the receiver, and the ability to process that data into a water vapour measurement. The first component has been successfully completed, the software package is capable of receiving raw GNSS measurement data (as well as some other necessary measurements such as approximate position and time) from the GNSS sensor and recording into a data file. The second component, the processing of the recorded data into a water vapour measurement is currently ongoing, and has raised a number of opportunities for further development of the system.

To quickly summarize the data processing work flow (described in previous deliverables): a number of factors shape raw GNSS measurements, the geometric distance between the GNSS satellite and the GNSS receiver, a number of "errors" (clock misalignments, etc.), and atmospheric conditions (including water vapour). A number of models are used to create initial "guesses" atmospheric conditions, then some state estimator algorithm (typically the extended Kalman filter) is used to compare the guesses with measurements, update the guesses accordingly, and recheck against new measurements until the guesses become consistent with all of the data, this gives the best estimate of atmospheric conditions. From this, water vapour amount can be easily calculated.

The current state of the data processing component is that it can read in the previously recorded data, model the initial "guesses" about atmospheric conditions, and has the infrastructure to do the recursive state estimation (though an exact algorithm has not been implemented yet). The major outstanding issues that need to be tackled to make the system capable of producing water vapour measurements are:

1. Implementation of the extended Kalman filter (EKF). The EKF is done with some linear algebra matrix decompositions, it is conceptually straight forward but tedious and time-consuming to implement.

2. The geometric distance between the receiver and GNSS satellites is calculated from ephemeris data, which are up-to-date descriptions of the satellite orbital parameters. This information is broadcast by the satellites themselves (very slowly), and is picked up by the receiver and stored on-board. The data-acquisition component is capable of querying this data from the receiver, but the ephemeris data to geometric distance transformation has not been implemented yet.

3. A number of small correction models need to be implemented: phase wind-up, solid earth tides, etc. Models for these exist in the literature but have not yet been included into my code.

Once the first two points, the Kalman filter and the geometric distance calculation, are implemented, the system will be capable of making rough water vapour measurements. The small corrections can be implemented after to improve accuracy. Once this point is reached, which is feasible over the course of the summer, then measurement validation can take place.

1.1.3 Measurement Validation

Proposed Milestone: A site where the GNSS measurements can be collected will be identified and a short campaign, spanning a number of days or weeks will be conducted to collect PWV measurements. A suitable source of PWV data for the local area will be identified to use as a "gold standard" against which to validate the system. The two will be compared and troubleshooting of the hardware and software components will be undertaken accordingly. Once validated, the comparison data will be prepared for publication.

Deliverable of this milestone would be a write-up of validation data and the methods used by the system. Goal: mid-May 2018.

Evaluation: Data collection and analysis can be easily automated with a simple script which causes data to be recorded periodically over the course of a few days. The data analysis component of the software is currently designed to read-in stored data and process it after the fact, so that can be easily done in batches after data collection has ceased. "Gold standard" data can be obtained from a number of sources, such as the ECMWF (European centre for medium-range weather forecasts) ERA-interim reanalysis dataset (http://apps.ecmwf.int/datasets/data/interim-full-daily/levtype=sfc/) after some amount of delay (can be up to 2-3 months, depending on data set). Data sets with this amount of time-delay will add an equivalent amount of time delay to the project, but should be manageable.

1.1.4 Hardware: Power Supply

Proposed Milestone: The requirements of a small solar power supply which could be used to make the unit autonomous and deployable will be determined. A power budget will be developed to determine the number of measurements which could be taken per day based on power consumption of the system and expected power generated per day. A scheme for handling the collected data would also be devised, either by transmitting it wirelessly or storing locally on flash memory.

Deliverable would be to have the system integrated with the solar power supply and the requisite control software developed to allow the system to operate within it's power budget.

1.1.5 Hardware: Weatherproofing

Proposed Milestone: The final milestone will be to enclose the moisture sensitive components of the system (the Raspberry Pi, power supply, and NEO-M8T breakout board) in a small, ruggedized hard case so that the system can be deployed outside. A thick case may negatively impact the performance of the GPS unit, if so, an active antenna can be attached directly to the NEO-M8T receiver and attached to the external portion of the case. Minor modifications may need to be made to the hard case to run an external power cable or active antenna lead through.

The deliverable will be a completed system which would be ready to be deployed. Further measurements could be collected with the completed system as proof of concept.

Evaluation: These power supply and weatherproofing have been given a much lower priority over getting the system working fully and validated, though may still be completed in the future.

2 Future Plans

The near-term goals of the project are to finish the implementation of the data-processing component of the software so that it can be used to generate water vapour measurements which can be used in the validation stage. However, given the time-delay to be expected in some of the "gold standard" data sets which will be used for validation, raw data could be collected much sooner and stored while the processing component is still being developed, and then analyzed when the "gold standard" validation data becomes available.

This is an advantage of being able to do the analysis as post-processing, however it also raises an interesting possibility of implementing the data-analysis in real-time with the data collection. This approach was taken by Yuan *et al.*, who made near-real-time estimates of water vapour by processing their data on-the-fly as they collected it [1]. A similar approach would be interesting to take with this system as it would make it much more useful for short-term and near-real-time weather prediction (nowcasting). The code is currently set up in such a way that this integration of the data processor with the data acquisition should be (relatively) smooth, though it will require the implementation of multithreading (essentially running the acquisition and processing as two concurrent programs). Conveniently, the *boost* libraries have solid support for multithreading.

A second avenue which could also be pursued is to investigate other state estimators than the extended Kalman filter (EKF). While the EKF is a tried-and-true algorithm for this problem, there has been a considerable amount of work in the field which could be investigated to find alternatives which may have advantages [2]. This would not only add scientific novelty, but also potentially improve the performance of the system.

Ultimately, what I would like to see at the conclusion of this project is a short technical note or scientific paper describing the system, demonstrating it's validation against canonical methods, and promoting the adoption of these very-low-cost sensors. Initial progress has been promising, the hardware has demonstrated that it is capable of taking raw GNSS measurements and the challenge lies now in putting all of the (software) pieces together to transform those into usable water vapour measurements. I have immensely enjoyed learning about GNSS systems while working on this project and greatly appreciate your support and feedback along the way.

References

[1] Yubin Yuan, Kefei Zhang, Witold Rohm, Suelynn Choy, Robert Norman, and Chuan-Sheng Wang. Real-time retrieval of precipitable water vapor from gps precise point positioning. *Journal of Geophysical Research: Atmospheres*, 119(16):10044–10057, 2014.

[2] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.