Thesis Script

1. Hello everyone and thank you for coming to my thesis proposal presentation. My name is Tommy Le, and my advisor is Dr. Ji-Chul Ryu. Today, I will be discussing my thesis titled “Magnetometer-less State-Estimation of a Mobile Robot Using Cascaded Kalman Filters”
2. Throughout this presentation, I will go over a brief introduction of localization and robotics, previous relevant work, the objective of this thesis, then will dive into the framework and working details of the proposed algorithm. Afterwards I will present my simulation results further solidifying the feasibility of the proposed algorithm, and then discuss the remaining work of the thesis.
3. State-estimation and localization is one of the fundamental blocks in the development of autonomous mobile robots. To provide robust effective navigation, a robust localization algorithm is needed. Most localization applications use an IMU along with an external reference sensor to provide a global measurement and compensate for the IMUs errors. For outdoor applications this sensor is typically a GPS. However for indoor applications, GPS signals are not accessible so implementations turn to magnetometers or UWB sensors. This thesis will focus on indoor localization and the shortcomings of using magnetometers for sensor fusion.
4. Before discussing further about the possible errors when using magnetometers, I would like to provide further background information on the inertial measurement unit or IMU. The IMU typically consists of a tri-axial accelerometer and gyroscope which is commonly paired with a magnetometer and are used to provide pose information. Although the IMU is the most common sensor used in localization and pose estimation, it has its own inherent errors. For example, the accelerometer contains high frequency noise which needs to be filtered out, the gyroscope suffers from drift in its measurements coming from the numerical integration of any DC bias that is in the signal, and the magnetometer is susceptible to hard and soft-iron distortion.
5. This leads me to the motivation for our thesis. Recently, our lab has been conducting research in the development and control of miniature spherical robots. One of the projects involved applications in the field of psychology for experiments with common lab animals, and an example image of it can be seen in the bottom left. A second version of the robot was created and is shown in the bottom right. One of the biggest challenges in the development of the spherical robot was designing a robot that could fit all our required hardware inside while remaining underneath a certain size. While eventually we were able to fit all the sensors within the robot’s shell, the magnetometer that was used suffered in performance due to its proximity to the motors and other electronics in the system. Possible solutions include adding a different sensor into the shell of the robot, which would increase its size past the required dimensions for our collaborator. Another solution would be to add an external sensor like a vision system or UWB tags, however this would require additional setup by the collaborator which is not ideal. This led to the idea of developing our own estimation algorithm that does not require additional sensors.
6. The area of localization for mobile robots is one that has been around for a long time. One project involves pose estimation for a robot using IMU and vision data. Here they use object recognition and feature matching to provide global pose information and fuse it with the IMU data to provide state estimation. For their EKF, the state transition matrices are the IMU and vision system and do not take into account the kinematic constraints of the mobile robot.
7. This work discusses the design and tracking of a spherical robot using IMUs and wheel odometry. For sensors, magnetometers, wheel encoders, and the 6 DOF IMU were used. The orientation estimation was handle through a complementary filter between the IMU and magnetometer and position data was estimated using dead reckoning by transforming the encoder data given the orientation data calculated using the complementary filter.
8. This paper titled “Enhanced 3D Kinematic Modeling of Wheeled Mobile Robots” proposes a kinematic model of a mobile robot that models the recursive velocity kinematics, wheel ground contact constraints, body-level slip prediction, liftoff prediction, and parameter calibration of the system. Although this algorithm was able to provide accurate trajectory tracking, the authors were not able to provide a real-time implementation due to the computational load this algorithm bears.
9. The final paper I would like to go over deals with pedestrian navigation which uses an IMU and magnetometer for pose estimation of a person walking. This paper was our main reference when designing our algorithm. Through multiple preprocessing methods and a cascaded Kalman filter framework, the authors of this paper were able to accurately estimate the position and orientation of pedestrian even in the presence of magnetic disturbance.
10. Overall, from literature review it was found that improvements in the field of localization and state-estimation can be made in the areas of magnetometer-less estimation. So, the objective of this thesis is to develop a robust magnetometer-less state estimation algorithm that should not require an additional sensor apart from an IMU. To compensate for the loss of a sensor, this project includes the kinematic constraints of the mobile robot to improve estimation. Although the goal of this project is to provide an algorithm for magnetometer-less state estimation, the option of using an additional sensor for improved localization should still be available if the use-case allows for it, therefore this thesis will also give the option to fuse an additional sensors data.
11. Before going into the details of the algorithm, it is necessary to provide a refresher into the Kalman Filter. The Kalman Filter is an optimal estimator used for linear systems. It follows the predict and update model where it will use a series of noisy measurements observed in time and produce estimates of unknown variables by keeping track of the estimated states of the system and uncertainty of those states. Some example applications of Kalman Filters and its variants include pose estimation, which will be the focus of this thesis, economics and estimating stock prices, as well as predicting the weather forecast.
12. Although, most systems that are dealt with in the real-world are not linear like the Kalman Filter assumes. This brings us to the Extended Kalman Filter. The EKF is practically an extension of the linear Kalman Filter for nonlinear systems, and linearizes our nonlinear model through the use of the Jacobian and Taylor expansions.
13. For complicated systems, one has the option to augment states from one subsystem to another or cascade Kalman Filters. The pros and cons of using a cascaded Kalman Filter instead are listed below and can be summarized as having a lower computational requirement and in turn a higher data rate and stronger real-time performance due to the simplicity of cascaded filters rather than large, augmented states.
14. Finally, this leads us to the overall framework of the algorithm. The algorithm contains the NMNI, NDZTA, and complementary filter preprocessing methods for improved pose information from IMU data, as well as the 1st EKF which estimates the course error of the mobile robot and the 2nd EKF which estimates the error in the states. The overall flow of the algorithm is as follows: first the sensor data has its bias removed using the bias estimates from the 2nd EKF and then is converted into pose data. The pose data then has its error corrected through the estimates in the 2nd EKF and is passed to the first EKF which uses the mobile robots kinematic model and compensated measurements to provide an optimal estimation of the mobile robot’s states.
15. The first preprocessing method I will discuss is related to the gyroscope called “No Motion No Integration.” This filters the gyroscope data by removing the bias and also creating a way to check if the gyroscope is actually in motion or not. The initial bias is removed by placing the IMU in a stationary position for a certain amount of time and finding the average value for the gyro readings, which is the bias of the gyro. This found bias is then subtracted from future gyro data to compensate for any drift in the gyro readings. During the sampling time the method also keeps track of the maximum gyro reading along each axis and stores that value as a threshold for motion. Depending on if the sensor reading is above or below that threshold, the filter can tell if the gyro is in fact in motion or not.
16. The next preprocessing method called “No Displacement Zero Translational Accumulation” is written by the same author and relates to the accelerometer. The methodology here is similar to the one mentioned previously. Here a low pass filter is applied to the accelerometer data, then sampling during a stationary period takes place and stores a threshold value that checks whether the IMU is in motion or not.
17. The final preprocessing method is the complementary filter. The complementary filter is a quick and effective sensor fusion method for blending accelerometer and gyroscope data. As recap an accelerometer does not accumulate any measurement bias like a gyroscope does but is noisy and a gyroscope provides accurate estimation over short periods of time but drifts over time. We desire the short-term accuracy of the gyro and long term stability of the accelerometer. By passing the orientation found from the accelerometer to a lowpass filter and the gyro orientation through a highpass filter, we are able to get the benefit of the gyro over short periods of time, and the stability of the accelerometer over long periods.
18. The first Kalman Filter in our algorithm is the course error correction block which uses the Unicycle model mobile robot as its process model. The unicycle model mobile robot has states x,y, and theta and control inputs v and w and the nonlinear model can be described by the middle equations. To use this process model with our desired EKF, the Jacobian of the model must be taken and is found as the matrix at the bottom right of the slide.
19. For this Kalman Filter, the initial measurements are acceleration and angular velocity, however we are externally converting our measurements to match the states of the robot through numerical integration and frame transformations which makes our measurement model an identity matrix and since the Jacobian of the identity matrix is itself, we are left with the matrix to the right as our measurement model for the 1st EKF.
20. The second Kalman Filter is the error state estimator which has orientation and velocity error as well as the gyroscope and accelerometer bias as its states. The input process noise vector w includes ita g and ita alpha representing white gaussian noise vectors for the gyroscopes and accelerometer as well as wg and wa which are the white gaussian noise processes for the sensor bias models. The change in orientation error is described by the summation of the transformed gyro bias and gyro noise. The change in velocity error is represented as S, the skew symmetric matrix of acceleration representing vector cross products, multiplied by the orientation error, summed with the transformed accelerometer bias and noise. That leaves the biases of the accelerometer and gyro which are modeled after the first order Gauss-Markov model where beta is the Markov process time constants that model our biases, and the omegas are the noise in our bias models.
21. The measurements for the 2nd EKF are the course error between the estimated heading from the 1st KF and the heading found through interpolation between the current and previous robot positions. The measurement model transforms all the orientation errors to be expressed in terms of heading error and has the 3x3 identity portion to describe the velocity measurements that is included in the states.
22. Before going into the simulation results, it is appropriate to mention how we generated our data and GT as well as why we decided to simulate our algorithm in the first place. The GT was created using a deterministic unicycle model mobile robots kinematics given a control input, v and omega. The desired trajectory for this simulation is a circular motion. The sensor data was then generated using the GT data and adding noise to the system. Additionally, random drift was added to the gyro to increase the accuracy of the simulation. Through accurately modeling our sensors, we hoped that the results would provide proof for the feasibility of our algorithm.
23. To illustrate the improvements in estimation using our proposed algorithm, I will be presenting the trajectory estimation in stages. The first presented is estimation using only the preprocessing methods and trapezoidal integration. As can be seen, the preprocessing methods alone do not work in giving robust pose estimation. This is due to the changing value of drift in the gyroscope and high amount of noise in the accelerometer data.
24. The next plot shows the 1st EKFs estimate with the unicycle model mobile robot process model. Here the estimation only seems to be off by a shift of 50 cm. This is most likely due to the presence of bias in the measurements and errors in the states.
25. Here we present the fully compensated trajectory estimation from the cascaded KF framework. The estimation is still slightly off and gives about +/- 15 cm accuracy in position however it can be seen that this algorithm provides a more accurate estimation than the previously mentioned while not requiring an additional sensor to provide additional information.
26. The remaining slides on the simulation results show how the states themselves compare to the ground truth generated by our kinematic model. The estimated heading is closely tracking the ground truth with a little noise in the estimate.
27. And the same can be said for the estimated velocities. Since these estimates are so close to the ground truth, it creates a bit of skepticism into our modeling of the systems noise and errors, but provides insight into the feasibility of the algorithm.
28. Now that the simulation is done, the remaining work for this thesis includes figuring out a fusion method for the incorporation of additional sensor data to provide improved accuracy in pose estimation, development of the algorithm in ROS Gazebo to allow for easily simulating the robot and transferring the program to a physical system.
29. The incorporation of an additional sensor is made an option so that users are able to use any data that may be readily available to improve the state estimation. For our application, the current idea is to use a vision system which will provide us position information, and from there we will be able to set up a cost function of squared errors and minimize it to find our ideal sensor information. From there the improved data will be passed to the course error correction EKF with the mobile robot process model to carryout the rest of the estimation.
30. In parallel, I will be working on developing the algorithm in ROS’s Gazebo simulator environment. Robot Operating System, or ROS, is a common robotics middleware used to interface common hardware components with software packages. With the addition of development in Gazebo, I am able to program the robot using its native programming language and package in a simulator, allowing for quick development without having to constantly upload a program and set up the robot for testing.
31. After the ROS Gazebo development is done, the program will be uploaded to the physical system where all the debugging and tuning of noise matrices will take place. Our robot for the experiment is going to be the Sphero-Mini. It is 40mm in diameter, has a 6DOF IMU embedded inside, and has a differential drive mechanism. The workspace for the robot will comprise of an overhead vision system to collect ground truth data on the robot as well as a piece of white poster board to improve color detection for tracking of the robot.
32. For testing the state estimation algorithm of the robot, we will use a vision-based control package that I developed last year in ROS to provide a desired trajectory and then in parallel will be estimating its pose information using our proposed algorithm. This video here is a demonstration of two robots interacting with each other using color detection for tracking and full-state feedback for control. For our application, one robot will be removed and the remaining robot will be given commands to follow our desired path.
33. With the implementation of our algorithm on the physical robotic system, we have a few potential challenges we would like to address. One would be the slip inherent in spherical robots. Typical mobile robots will have wheel slip between the ground and point of contact on the wheel. A spherical robot will have slippage between the wheel and shell of the robot, as well as the shell and ground contact point causing noisy or erroneous measurements. To accommodate for this, we believe it may be necessary to change the workspace of the robot to a platform with a larger coefficient of friction so that the slip will be lessened. Currently on the posterboard, the robot is prone to slip but with slower speeds and a rougher surface, the slip should be reduced. The limit on the speed of the robot should also address the difficulties in interpolated heading error a well as the issue with ROS’s real-time operating abilities. Limiting the speed of the robot should lower the distance traveled over time, and with the distance traveled over time lessened the refresh rate need to accurately capture the robot’s data is lessened as well.
34. With all aspects of my project being addressed, here is a gantt chart showing the remaining work and timeline of my thesis. After this proposal, I will begin development of my algorithm in ROS and finish developing the path planning portion of the project. That should be done around the time winter break starts. Then during winter break, I will begin testing and debugging the algorithm on the physical system while in parallel working on implementing the method for incorporating the additional sensor data. Then come spring semester, I am expecting to finish up testing and data acquisition and analysis and begin working on documentation and my thesis defense.