UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Unscented Kalman Filter Highway Project

REVIEW

CODE REVIEW 3

HISTORY

Meets Specifications

Keen Learner,
Congratulations!🎉🎉
This implementation shows a good understanding of the underlying UKF concepts and I commend you for the neat and organized code in this submission. From the RMSE values, you can clearly see the improvement that the UKF brings compared to the EKF. Stay focused and hard working as always and all the best moving forward!

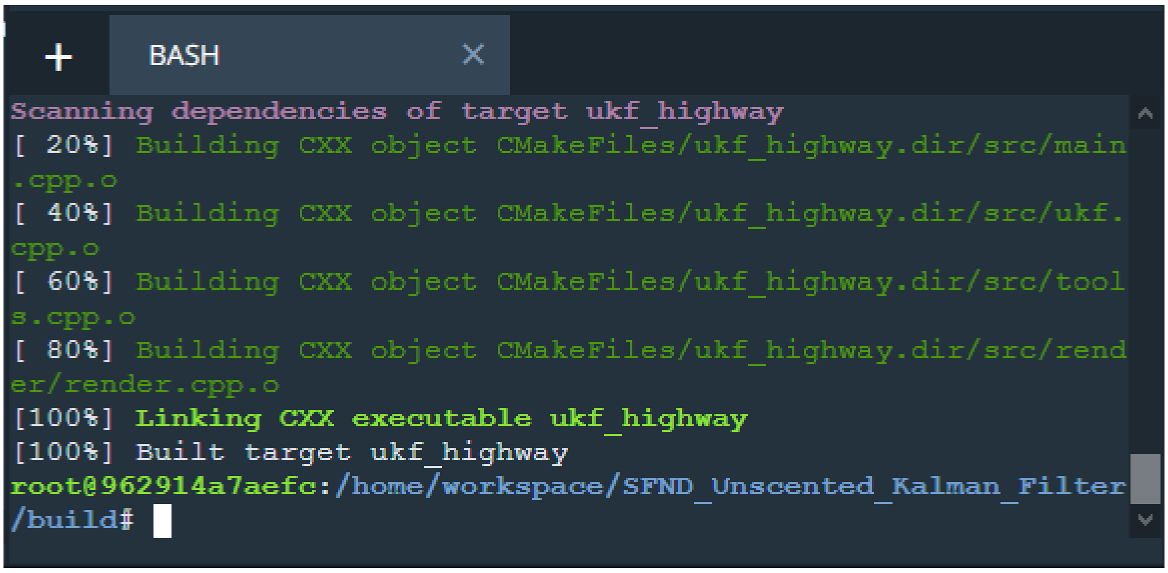
Advance Learning Tips
Here are a few resources you might want to refer to for more information on Unscented Kalman Filters.

The Unscented Kalman Filter for Nonlinear Estimation
Unscented Kalman Filter Tutorial
Uncented Kalman Filter for Dummies
Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter

Compiling and Testing

The project code must compile without errors using `cmake` and `make`.

Upon compiling the code using cmake and make, no errors were found as the compilation was 100% successful. Well done!



```
Scanning dependencies of target ukf_highway
[ 20%] Building CXX object CMakeFiles/ukf_highway.dir/src/main
.cpp.o
[ 40%] Building CXX object CMakeFiles/ukf_highway.dir/src/ukf.
cpp.o
[ 60%] Building CXX object CMakeFiles/ukf_highway.dir/src/tool
s.cpp.o
[ 80%] Building CXX object CMakeFiles/ukf_highway.dir/src/rend
er/render.cpp.o
[100%] Linking CXX executable ukf_highway
[100%] Built target ukf_highway
root@962914a7aefc:/home/workspace/SFND_Unscented_Kalman_Filter
/build#
```

Advance Learning Tips:

Debugging Strategies, Tips, and Gotchas,
Techniques for Debugging in C++,
Some favorite debugging techniques in C++ as discussed on stack overflow.

Code Efficiency

Your code does not need to sacrifice comprehension, stability, or robustness for speed. However, you should maintain good and efficient coding practices when writing your functions.

Here are some things to avoid. This is not a complete list, but there are a few examples of inefficiencies.

- **Running the exact same calculation repeatedly when you can run it once, store the value and then reuse the value later.**
- **Loops that run too many times.**
- **Creating unnecessarily complex data structures when simpler structures work equivalently.**
- **Unnecessary control flow checks.**

This Implementation stays clear of unnecessary complex data structures, avoids loops that run repeatedly, avoids running exact calculations severally and also unnecessary control flow checks are avoided.

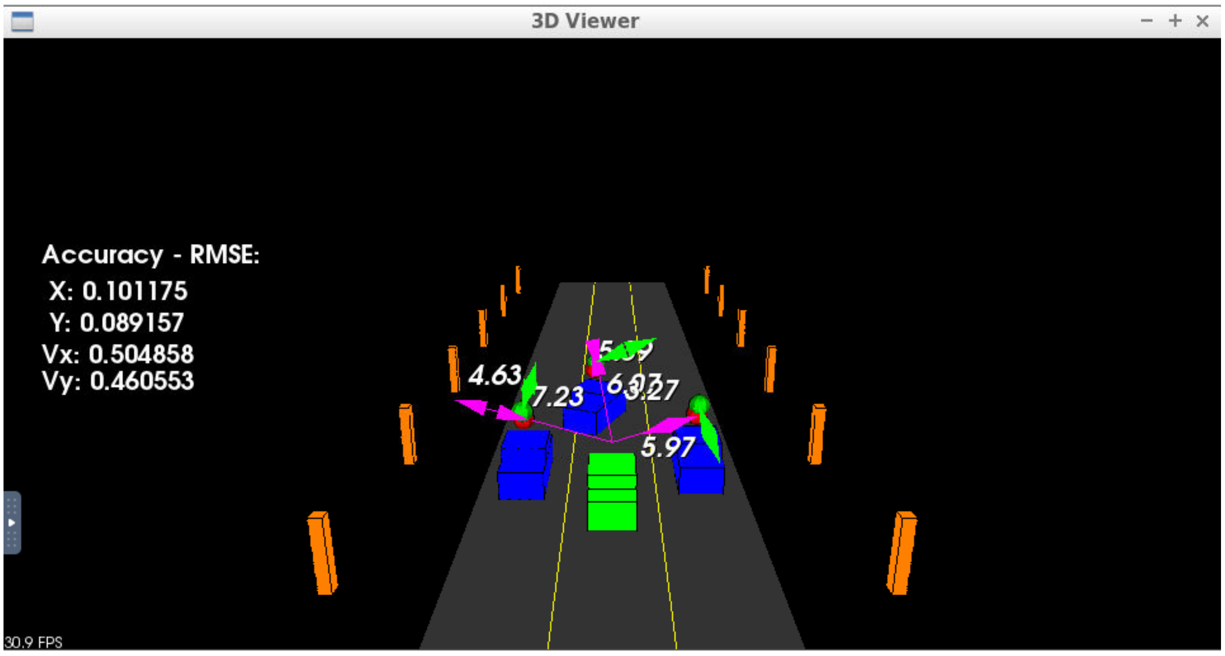
Advance Learning Tips:
Here are a few tips for improving code efficiency and optimization:

Optimizing C++/Writing efficient code/Performance improving features.
Efficient C++ Performance Programming Techniques
10 Tips for C and C++ Performance Improvement Code Optimization

Accuracy

The simulation collects the position and velocity values that your algorithm outputs and they are compare to the ground truth data. Your px, py, vx, and vy RMSE should be less than or equal to the values [0.30, 0.16, 0.95, 0.70] after the simulator has ran for longer than 1 second. The simulator will also display if RMSE values surpass the threshold.

The results are good. The RMSE values are well within the required range. This only goes to show that the algorithm used was carefully implemented and all necessary caution was taken. This is really commendable!
✅'



```
Accuracy - RMSE:
 X: 0.101175
 Y: 0.089157
Vx: 0.504858
Vy: 0.460553
```

Follows the Correct Algorithm

While you may be creative with your implementation, there is a well-defined set of steps that must take place in order to successfully build a Kalman Filter. As such, your project should follow the algorithm as described in the preceding lesson.

This submission adequately follows the predefined steps from the lessons. A good job is done in ukf.cpp, implementing all the computational steps required to achieve a successful Unscented Kalman Filter with clearly written codes. This I must say was nicely done!

DOWNLOAD PROJECT

3 CODE REVIEW COMMENTS ›