**Assignment 4: Kalman Filters**
**CS 393R: Autonomous Robots**
**Due Date: 11:59:59 PM Wednesday, October 10, 2018**

Your task is to design a Kalman Filter to help your robot track the ball, compute its velocity, and anticipate shots on goal. You will design a simple goalie behavior that indicates when the ball is approaching and toward which side of the goalie.

First, you will write a Kalman Filter in C++ using only the C++ Standard Library and a linear algebra library of your choice. Eigen is strongly encouraged, but other libraries are available in the codebase, and any library you find online is acceptable so long as it doesn't implement the Kalman Filter itself. Your implementation should not impose any practical limits on the dimensionality of your state or measurement models. Furthermore your implementation should deduce and apply all matrix and vector dimensions at compile time.

Once you have programmed a generalized filter you will need to design state and measurement models for the orange ball, including dimensionalities, update transforms, and uncertainties. You can incorporate any sensors in the robot you like, directly or indirectly, in whatever coordinate frames work best. You may tune your covariance matrices by hand or analytically so long as you do all of the analysis with your own code. Again, you are restricted to Standard Library and Linear Algebra libraries for this component.

When your filter is ready you must use it as the basis for a goalie behavior that visually indicates when and where the ball is being shot. To avoid damaging the robot, we will not be allowing physical movement for this assignment other than adjusting the arms. Instead, you may raise the arms in the direction that the ball is going. The arms should be straight for all three indicators. When indicating left, raise the left arm so that it is pointing in the *+y* direction. For the right side, point the right arm in the *-y* direction. For the center, point both arms in the *+x* direction. The arms must be parallel to the ground before the ball passes or contacts the robot.

You may find it helpful to test your Kalman filter using the goalie simulator. To access the environment, start up the tool, open the World window, then select "GoalieSim" from the dropdown. The simulator will play your default behavior which is defined by the call to `runBehavior` in `init.py`. Your goalie will receive noisy ball position data through the `WorldObjectBlock`, which it will need to filter and act upon in the simulator. This data can be processed through the localization module just as it is on the physical robot.

The file `core/localization/LocalizationModule.cpp` has some sample code to get you started. Most of this code revolves around taking vision data and populating memory blocks with processed data so that the results of your filter can be visualized in the tool and used by the robot. The file `behaviors/keeper.py` has a simplistic goalie behavior that you can work from as well.

**Checklist:**

1. [____] (2 points) Demonstrate that your Kalman Filter implementation can model states and measurements of arbitrary dimensionality (1 point) and that matrix and vector sizes are defined at compile time (1 point). Passing a compile-time parameter into a dynamically sized matrix or vector object is not sufficient.

2. [____] (6 points) Demonstrate goalie's ability to predict and indicate whether the ball is approaching its left, center, or right on the field. There will be exactly 2 tries each direction, 1 point each. Each false positive will reduce your score by 1 point, up to 2 points per side.

3. [____] (4 points) Demonstrate goalie's ability to avoid false positives. A stationary ball will be placed at different points near the robot. The ball will also be rolled in front of the robot such that there is no chance of the ball going into the goal. You will lose 1 point for each block executed up to a maximum of 4.

4. [____] (2 points) Clarity and quality of your memo. Email it (along with a compressed folder of your code) to Peter and Josiah by the due date. At the top of your memo **please clearly indicate where your Kalman Filter implementation can be found**.

**Extra Credit:**

5. [____] Implement the Extended Kalman Filter (2.5 points) as described in Probabilistic Robotics, Chapter 3. You must use this implementation for the main components of the assignment.

**Initials:** _____ TA _____ _____ Teammates _____