Roll-Pitch-Yaw Demo Firmware

W. Premerlani

The roll-pitch-yaw demo showcases the capabilities of the “Rmatrix direction cosine” firmware, recently completed by the author with the help of Paul Bizard. If you have a UAV development board, this demo firmware is a good way for you gauge the high performance that the Rmatrix algorithm achieves in combining GPS, gyro, and accelerometer information into orientation information. What the demo does is to continually compute the 9 elements of the direction cosine matrix, and display 3 of them via servo outputs. Servo 1 displays the cosine of the angle between the X axis of the board and the local earth Z axis. Servo 2 displays the cosine of the angle between the Y axis of the board and the local earth Z axis. Servo 3 displays the cosine of the angle between the Y axis of the board and the local earth Y axis. The author normally mounts his board in his aircraft with the Y axis board label pointing forward, Z axis down, (components facing up) and the X axis completing a right handed coordinate system. Servo 1 is indicative of roll, servo 2 indicates pitch, and servo 3 indicates yaw.

To try out the firmware, after you compile it and program your board, do the following:

1. Connect 3 servos to the 3 servo outputs of the board.
2. Do *NOT* connect a GPS radio to the board at first, it is interesting to see how the firmware behaves without yaw information from the GPS.
3. Place the board on an approximately level surface, components facing up. This is important for zeroing of the accelerometers.
4. Power up the board. The author does this by connecting a 4.8 NiCad battery directly to one of the radio input channels on the board. Be careful to get minus on the pin closest to the edge of the board, and plus on the center pin.
5. While the board is powering up the gyros and accelerometers are being zeroed, so it is important for the board to remain absolutely motionless. If you accidentally move the board, reset it. During this time one of the LEDs will be lit.
6. After the zeroing is complete, the LED will turn off and one of the servos will deflect.

After the board powers up, the servos will be practically motionless, until you move the board, in which case the three servos will indicate roll, pitch, and yaw. Here are a few things you might want to try that were also previously published as the roll-pitch demo experiments.

1. Starting with the board level, slowly rotate it a little bit around each of its axes, one at a time. You will know that the rotation is slow enough if the LED does not come on. Notice that, with the board level, rotation around the Z axis produces very little servo deflection for servos 1 and 2 and that they have

zero drift. This is because the small amount of residual gyro drift in roll and pitch is completely compensated for using accelerometer information. There is drift in servo 3, the yaw servo, because the GPS is not connected yet. However, notice that the drift is hardly noticeable.

2. Maintaining a level orientation of the board, shake it, and notice very little servo deflection. *This is because nearly all of the orientation information comes from the gyros*.
3. Slowly turn the board upside down, and observe the servos. Slowly return it to right side up, and observe that the servos return to their starting positions.
4. Slowly rotate the board so that its X axis is vertical. Rotate the board a little bit around each of its axes, one at a time, and observe the response of the servos.
5. Rotate the board rapidly and notice that the LED comes on. Depending on which way you rotated it, the servos may slowly return to their correct positions.

The next set of experiments you will demonstrate yaw performance by taking the board along for a walk. For what is to follow, the author found it convenient of Velcro all components to a thick piece of cardboard. You might want to do something similar.

1. Connect the GPS to the board and take everything outside. Set the board on a level surface with the Y arrow of the board facing north, and press the reset button. Hold the board level and motionless until the LED goes out. Then, wait for the GPS to synchronize, if it has not done so already. When the board has valid navigation information from the GPS, an LED on the nav board will flash once a second. The board is now ready for a walk. (By the way, this firmware uses the binary interface to the GPS. As a result, the LED on the GPS will stop flashing. Do not worry, that is normal.)
2. Go for a long walk at a normal pace, with the Y arrow facing the direction that you are moving. *It is important that you keep the Y arrow facing the approximate direction that you are traveling, because that is what it will do in a plane, and that is the underlying assumption of the yaw drift correction.* Make turns at your pleasure, but try not to exceed a rotation rate of 75 degrees per second. If you do, the LED will produce additional flashes to indicate gyro saturation. As you turn, servo 3 will indicate what direction you are walking. Keep in mind that it is displaying the direction cosine, which will be +1 when you are moving north, -1 when you are moving south, and 0 when you are going east or west.
3. If you would like to explore the initial locking behavior of the algorithm, reset the board while it is facing east (or west). In effect, you are creating a 90 degree yaw error in the Rmatrix. Start walking, and you will observe the yaw error gradually decaying.
4. Walk in a straight line for a couple of minutes. Stop. Rotate the board 360 degrees about its Z axis. You will observe servo 3 indicate a complete 360 degree rotation. Resume your walk.

5. Make some extremely fast turns to saturate the gyros. (This might not be easy to do.) This will induce some yaw error. Continue your walk and observe the error decay.
6. During your walk, the firmware will be continuously adjusting the offsets of the gyros. After a few minutes, the offsets will be completely removed. Stop, so that the GPS direction information becomes invalid. Notice how stable the yaw behavior is even with erroneous GPS direction information.

The roll-pitch-yaw demo firmware uses GPS, gyros and accelerometers to maintain a direction cosine matrix that describes the orientation of the board with respect to the local earth frame of reference. There are nine direction cosine elements arranged in a three by three matrix. Each element is the cosine of the angle between one of the axes of the board and one of the axes of the local earth frame of reference. The reference axes of the board are labeled on the board. The Z axis is down.

The cosine matrix is used to transform vectors back and forth between the board and the local earth frame of reference. It has the interesting property that its inverse is equal to its transpose. It is all you need to represent the orientation of the board, and can be used to provide a basis for an accurate and fast rudder, aileron, and elevator control that will work rather well under any and all conditions, including sideways and upside-down.

The roll-pitch-yaw demo is the completion of the firmware implementation of the algorithm developed by the author and Paul Bizard for maintaining an accurate estimate of the direction cosine matrix, which we call the Rmatrix algorithm. We both worked on theory, Paul is also doing simulations, and the author is doing firmware implementation. The firmware is complete. The following is executed approximately every 0.025 seconds.

1. Update the direction cosine matrix by numerically integrating the effects of rotation. This can be done rather efficiently by using a MicroChip matrix multiply routine that is tailored to the dsPIC30.
2. Recognizing that accumulated numerical errors will violate the orthogonality property of the matrix, renormalize the matrix.
3. Account for centrifugal effects by taking the cross product of the speed vector and the rotation rate vector.
4. Compare the gravity vector measured by the accelerometers to the last row of the matrix. Use the discrepancy to compensate for drift in the gyros. This will keep roll-pitch error from accumulating.
5. Use information from the GPS to compensate for yaw drift.

We considered using a regression analysis to fine tune the gyro gains, but it turns out that the algorithm is sufficiently robust to work with nominal gyro gains.

More firmware and papers are planned. Eventually there will be a paper describing the theory, and probably a paper presenting simulation results. The next two planned firmware releases are:

1. Revision of the GentleNav firmware to use the direction cosine matrix for pitch and yaw control. This is expected to produce performance that is an

order of magnitude more precise, responsive, and stable than the existing version.

2. Firmware that will implement the features of the GentleNav firmware for aircraft with ailerons for full pitch, yaw, and roll control.

Releases will be published as soon as they have been simulated and tested on the ground. Flight testing will be done in the spring.