Estimation Project

1. Update From IMU

```
///////////////////////////// BEGIN STUDENT CODE ////////////////////////////
// SMALL ANGLE GYRO INTEGRATION:
// (replace the code below)
// make sure you comment it out when you add your own code -- otherwise e.g. you might integrate yaw twice

Quaternion<float> UpdateIMU = Quaternion<float>::FromEuler123_RPY(rollEst, pitchEst, ekfState(6));

UpdateIMU.IntegrateBodyRate(gyro,dtIMU);


float predictedPitch = UpdateIMU.Pitch();
float predictedRoll = UpdateIMU.Roll();
ekfState(6) = UpdateIMU.Yaw();    // yaw

// normalize yaw to -pi .. pi
if (ekfState(6) > F_PI) ekfState(6) -= 2.f*F_PI;
if (ekfState(6) < -F_PI) ekfState(6) += 2.f*F_PI;

///////////////////////////// END STUDENT CODE ////////////////////////////
```

2. Predict State

```
////////////////////////////// BEGIN STUDENT CODE ///////////////////////////

predictedState(0) = curState(0) + dt * curState(3);
predictedState(1) = curState(1) + dt * curState(4);
predictedState(2) = curState(2) + dt * curState(5);

V3F accel_dot = attitude.Rotate_BtoI(accel);

predictedState(3) = curState(3) + dt * accel_dot.x;
predictedState(4) = curState(4) + dt * accel_dot.y;
predictedState(5) = curState(5) + dt * accel_dot.z - 9.81*dt;

////////////////////////////// END STUDENT CODE ///////////////////////////
```

3. Get Rbg Prime
From estimation of Quadrotor material eq.(52)

```
////////////////////////////// BEGIN STUDENT CODE ////////////////////////////
float theta = pitch;
float phi = roll;
float psi = yaw;

RbgPrime(0, 0) = -cos(theta) * sin(psi);
RbgPrime(0, 1) = -(sin(phi) * sin(theta) * cos(psi)) - (cos(phi) * cos(psi));
RbgPrime(0, 2) = -(cos(psi) * sin(theta) * sin(psi)) + (sin(phi) * cos(psi));
RbgPrime(1, 0) = cos(theta) * cos(psi);
RbgPrime(1, 1) = sin(phi) * sin(theta) * cos(psi) - (cos(theta) * sin(psi));
RbgPrime(1, 2) = cos(phi) * sin(theta) * cos(psi) + sin(phi) * sin(psi);
RbgPrime(2, 0) = 0;
RbgPrime(2, 1) = 0;
RbgPrime(2, 2) = 0;

////////////////////////////// END STUDENT CODE ////////////////////////////
```

4. Predict
From estimation of Quadrotor material eq.(50)

```
////////////////////////////// BEGIN STUDENT CODE ////////////////////////////
gPrime(0, 3) = dt;
gPrime(0, 4) = dt;
gPrime(0, 5) = dt;
gPrime(3, 6) = (RbgPrime(0) * accel).sum() * dt;
gPrime(4, 6) = (RbgPrime(1) * accel).sum() * dt;
gPrime(5, 6) = (RbgPrime(2) * accel).sum() * dt;

ekfCov = gPrime * ekfCov * gPrime.transpose() + Q;


////////////////////////////// END STUDENT CODE ////////////////////////////
```

5. Update From GPS

```
////////////////////////////// BEGIN STUDENT CODE ////////////////////////////
for (int i = 0; i < 6; i++) {
    hPrime(i, i) = 1;
}
////////////////////////////// END STUDENT CODE ////////////////////////////
```

6. Update From Magnetometer

```
////////////////////////////// BEGIN STUDENT CODE //////////////////////////
hPrime(0, 6) = 1;
zFromX(0) = ekfState(6);
float correction = z(0) - zFromX(0);
if (correction > F_PI) {
    zFromX(0) += 2.f * F_PI;
}
else if (correction < F_PI) {
    zFromX(0) -= 2.f * F_PI;
}

////////////////////////////// END STUDENT CODE ////////////////////////////
```