```python
"""
Homework 1
Math 404
1/20/16

@author: Jessica Morrise
"""

import numpy as np
import scipy.linalg as la

def data_generator(n_steps):
    """
    Generate some kind of data.
    Yield A, b and R for each timestep.
    """
    m = 8
    n = 2
    A = np.random.rand(m,n)*5 + 5
    for i in xrange(n_steps):
        A = A + np.random.rand(m,n)*0.05
        b = np.random.rand(m,1) + i
        R = np.random.rand(m,m)
        yield A,b,R

def RLS_step(K_1, A, b, R, x_hat_1):
    """
    Given A, b, R, and the previous x_hat and K, estimate the
    state at the current step using RLS.
    """
    #Compute K
    inv_matrix = la.inv(R + np.dot(A,K_1).dot(A.T))
    K = np.dot(K_1,A.T).dot(inv_matrix).dot(A).dot(K_1)
    K = K_1 - K
    #Compute x_hat
    R_inv = la.inv(R)
    x_hat = np.dot(K,A.T).dot(R_inv).dot(A.dot(x_hat_1) - b)
    x_hat = x_hat_1 - x_hat

    return x_hat, K

def OLS(A,b,W):
    """
    Solve the weighted least squares problem
    """
    AW = np.dot(A.T, W)
    x_hat = la.solve(AW.dot(A), AW.dot(b))
    return x_hat

def RLS(num_steps = 20):
    """
    Solve the RLS problem using the generated data.
    Simultaneously solve using OLS in order to compare the results.
    """
    k = 0
    estimates = []

    for Ak, bk, Rk in data_generator(num_steps):
        if k == 0:
            #Set up RLS problem
            x_hat = OLS(Ak,bk,Rk)
            K = la.inv(np.dot(Ak.T,la.inv(Rk)).dot(Ak))
            #Set up OLS problem
            x_hat_ols = x_hat.copy()
            A_ols = Ak
            b_ols = bk
            W_ols = la.inv(Rk)
    else:
        #Solve RLS problem
        x_hat, K = RLS_step(K,Ak,bk,Rk,x_hat)
        #Solve OLS problem
        A_ols = np.vstack((A_ols, Ak))
        b_ols = np.vstack((b_ols, bk))
        W_ols = la.block_diag(W_ols, la.inv(Rk))
        x_hat_ols = OLS(A_ols, b_ols, W_ols)

    print "RLS:",x_hat.T
    print "OLS:",x_hat_ols.T, '\n'
    estimates.append(x_hat)
    k += 1
return estimates
```