```
function [g_YX,g_WZ,g_YZ,g_XW] = correlators3(lambda,phi,Tmax,cycles,quantum_efficiency,dark_rate,blur,dr,phys,Y_bias,X_bias)
    % correlators3 calculates the individual g2 correlators using the
    % back-to-back histograms rather than standard histograms.  It utilises
    % mypdist2 which will return the separated histograms for each
    % port-pair.
    colist_test = coordinatelist(lambda,phi,Tmax,cycles,quantum_efficiency,dark_rate,blur,phys,Y_bias,X_bias);

    % Start with empty distance lists.
    YXdlist = [];
    WZdlist = [];
    YZdlist = [];
    XWdlist = [];
    % Within a cycle numerator for our g2 correlators.
    for i = 1:cycles
        temp = colist_test{i};
        % Record the distances of pairs within each cycle.
        [holdYX,holdYZ,holdXW,holdWZ] = mypdist2(temp);
        YXdlist = [YXdlist holdYX];
        WZdlist = [WZdlist holdWZ];
        YZdlist = [YZdlist holdYZ];
        XWdlist = [XWdlist holdXW];
    end

    % Combine the cycles together to find the independent averages or
    % denominator for g2.
    colist2 = [];
    for i=1:cycles
        colist2 = [colist2; colist_test{i}];
    end
    [YXd2list,YZd2list,XWd2list,WZd2list] = mypdist2(colist2);
    dr2 = 1+sqrt(1-(dr^2)/4);
    edges = 0:dr:2.1;

    % Calculate the g2 numerator/denominator for each port pair
    h1_YX = histogram(YXdlist,edges);
    f_YX = h1_YX.Values;
    h2_YX = histogram(YXd2list,edges);
    s_YX = h2_YX.Values;


    h1_WZ = histogram(WZdlist,edges);
    f_WZ = h1_WZ.Values;
    h2_WZ = histogram(WZd2list,edges);
    s_WZ = h2_WZ.Values;

    h1_YZ = histogram(YZdlist,edges);
    f_YZ = h1_YZ.Values;
    h2_YZ = histogram(YZd2list,edges);
    s_YZ = h2_YZ.Values;
```

```
    h1_XW = histogram(XWdlist,edges);
    f_XW = h1_XW.Values;
    h2_XW = histogram(XWd2list,edges);
    s_XW = h2_XW.Values;

    % Find the g2 correlators.
    fsum = sum(f_YX)+sum(f_WZ)+sum(f_YZ)+sum(f_XW);
    ssum = sum(s_YX)+sum(s_WZ)+sum(s_YZ)+sum(s_XW);

    gYX = (f_YX./fsum)./(s_YX./ssum);
    g_YX = gYX(1);

    gWZ = (f_WZ./fsum)./(s_WZ./ssum);
    g_WZ = gWZ(1);

    gYZ = (f_YZ./fsum)./(s_YZ./ssum);
    g_YZ = gYZ(1);

    gXW = (f_XW./fsum)./(s_XW./ssum);
    g_XW = gXW(1);
end
```