

---

```

function [ predX, predY ] = predictLocation( maxResponseX,
maxResponseY, frameLookbacks )

    numDataPoints = size(maxResponseX, 1);
    dims = ceil(numDataPoints / 2);

    % Hx is the Hankel matrix for the x response values
    Hx = zeros(dims, dims);

    % Hy is the Hankel matrix for the y response values
    Hy = zeros(dims, dims);

    for i = 1:dims
        for j = 1:dims
            cellNum = i + j - 1;
            Hx(i, j) = maxResponseX(cellNum);
            Hy(i, j) = maxResponseY(cellNum);
        end
    end

    % all but the last column of the H matrix is A
    Ax = Hx(:, 1:dims-1);
    Ay = Hy(:, 1:dims-1);

    % the last column of the H matrix is b
    bx = Hx(:, dims);
    by = Hy(:, dims);

    vx = Ax \ bx;
    vy = Ay \ by;

    cLength = dims-1;
    Cx = zeros(1, cLength);
    Cy = zeros(1, cLength);

    % Cx and Cy are used with vx and vy to predict a new X and Y
    for i = 1:cLength
        Cx(1, i) = maxResponseX(numDataPoints - (dims - i) + 1);
        Cy(1, i) = maxResponseY(numDataPoints - (dims - i) + 1);
    end

    % This iterates once for each future prediction to be made
    for i = 0:frameLookbacks
        predX = Cx * vx;
        predY = Cy * vy;

        % We want to shift the Cx and Cy arrays left and add the
        % newly-predicted X and Y's in case we want to predict X and Y
        % again
        Cx = circshift(Cx(:, 1:cLength), [0 cLength-1]);
        Cy = circshift(Cy(:, 1:cLength), [0 cLength-1]);
        Cx(:, cLength) = predX;
    end

```

---

---

```
        Cy(:, cLength) = predY;  
    end  
end
```

*Published with MATLAB® R2017a*