```
function [t_out, x_out, P_out, nis] = kalman_filter(F, G, Gamma, H, Q, R, u,
z, xhat0, P0)
%KALMAN_FILTER The most basic KF ever to exist

% Should update this to be able to accept a G and u

    num_meas = size(z, 1)/size(H, 1);
    nx = length(xhat0);

    % Initialize output
    t_out = [0 repelem(1:num_meas, 2)].';
    x_out = zeros(num_meas*2 + 1, length(xhat0));
    x_out(1, :) = xhat0;
    P_out = zeros([size(P0), num_meas*2 + 1]);
    P_out(:, :, 1) = P0;
    if nargout > 3
        nis = zeros(num_meas, 1);
    end

    % Inititialize priors
    x_post = xhat0;
    P_post = P0;

    % Recursive Estimation
    for k=1:num_meas

        % Prediction step
        x_prior = F*x_post + G*u(k, :).';
        P_prior = F*P_post*F.' + Gamma*Q*Gamma.';

        % Store the prediction
        x_out(2*k, :) = x_prior;
        P_out(:, :, 2*k) = P_prior;

        % Update step
        nu = z(k, :).' - H*x_prior;
        S = H*P_prior*H.' + R;
        K = P_prior*H.'/S;
        x_post = x_prior + K*nu;
        P_post = (eye(nx) - K*H)*P_prior*(eye(nx) - K*H).' + K*R*K.'; %
Joseph form for numerical stability

        % Store the correction
        x_out(2*k + 1, :) = x_post;
        P_out(:, :, 2*k + 1) = P_post;

        if nargout > 3
            nis(k) = nu.'*inv(S)*nu;
        end
    end
```

```
end
```

Published with MATLAB® R2025a