

---

```

function [t_out, x_out, P_out, nis] = srif(F, Gamma, H, Q, R, xhat0, P0, z)
%SRIF Square-root information filter for analysis (not real time, since we
% are transforming back to conventional representation)

% Characterize the data
nx = length(xhat0);
nz = size(z, 2);
nv = size(Q, 1);
num_meas = size(z, 1)/size(H, 1);

% Initialize outputs
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

% Compute some constants
Ra = chol(R); % This is not necessarily the same for every measurement in
general
Rait = inv(Ra).';
G = zeros(nx, 1); % No inputs for now
u = 0; % No inputs for now

% Set up initial priors
Rxx = inv(chol(P0)).'; % Smarter way to do this in the notes
Rvv = inv(chol(Q)).';
vhat = zeros(size(Q, 1), 1);
zv = Rvv*vhat;
zx = Rxx*xhat0;

for k=1:num_meas

    % Propagate
    lower_left = -Rxx*(F\Gamma);
    lower_right = Rxx/F;
    [Qb, Rb] = qr([Rvv, zeros(nv, nx); lower_left, lower_right]);

    bottom = zx + Rxx*(F\G)*u;
    zb = Qb.*[zeros(nv, 1); bottom];

    % Extract block elements from matrices
    zx_bar = zb(end-nx+1:end); % I hate matlab indexing
    Rxx_bar = Rb(end-nx+1:end, end-nx+1:end);

    % Save P_bar and x_bar
    x_out(2*k, :) = Rxx_bar\zx_bar;
    P_out(:, :, 2*k) = inv(Rxx_bar)*inv(Rxx_bar).';

end

```

---

---

```
% Update
zk = z(k, :).';
za = Rait*zk;
Ha = Rait*H;

[Qc, Rc] = qr([Rxx_bar; Ha]);
zc = Qc.'*[zx_bar; za];

% Extract block elements from matrices
Rxx = Rc(1:nx, 1:nx);
zx = zc(1:nx);

% Compute NIS
if nargout > 3
    zr = zc(end-nz+1:end);
    nis(k) = zr.'*zr;
end

% Save P and x_hat
x_out(2*k + 1, :) = Rxx\zx;
P_out(:, :, 2*k + 1) = inv(Rxx)*inv(Rxx).';

end

end
```

*Published with MATLAB® R2025a*