```
function f = uni_ekf_test2
% EN530.603 Extended Kalman filtering of the unicycle with bearing and range measurements
%
% M. Kobilarov , marin(at)jhu.edu

clear

%rng('default')
rng(10212)

S.bearing_only = 1;

% single beacon at (-2,2) : system is unobservable
%S.pbs = [-2;
%         2];    % beacon positions

% two beacons at (-2,2) and (2,2) : system is observable (two or more)
S.pbs = [-2, 2;
         2, 2];    % beacon positions

nb = size(S.pbs,2); % number of beacons

if S.bearing_only
  S.h = @b_h;    % bearing sensing
  S.r = nb;      % measurement dimension
  S.R = .4*diag(repmat([.1], nb, 1));
else
  S.h = @br_h;   % bearing-reange sensing
  S.r = 2*nb;      % measurement dimension
  S.R = .4*diag(repmat([.1; .01], nb, 1));
end

S.n = 4;       % state dimension
S.f = @uni_f;  % mobile-robot dynamics

% timing
dt = .1;
%N = 2580;
N = 50;
T = dt*N;
S.dt = dt;

% noise models
S.Q = dt^2*diag([.01 .01 .01 .0001]);


% initial mean and covariance
xt = [0; 0; 0; 1]; % true state

P = diag([.01 .01 .01 .04]); % initial covariance
x = xt + sqrt(P)*randn(S.n, 1); % initial estimate with added noise

xts = zeros(S.n, N+1); % true states
xs = zeros(S.n, N+1);  % estimated states
Ps = zeros(S.n, S.n, N+1); % estimated covariances
ts = zeros(N+1,1); % times

zs = zeros(S.r, N);  % measurements
```

```
xts(:, 1) = xt;
xs(:, 1) = x;
Ps(:, :, 1) = P;
ts(1) = 0;

ds = zeros(S.n, N+1);  % errors
ds(:,1) = x - xt;

for k=1:N
  u = dt*[2; 1];  % known controls

  xts(:,k+1) = S.f(xts(:,k), u, S) + sqrt(S.Q)*randn(4,1);  % true state

  [x,P] = ekf_predict(x, P, u, S);  % predict
  ts(k+1) = k*dt;

  z = S.h(xts(:,k+1), S) + sqrt(S.R)*randn(S.r,1);  % generate measurement
  [x,P] = ekf_correct(x, P, z, S);  % correct

  xs(:,k+1) = x;
  Ps(:,:,k+1) = P;

  zs(:,k) = z;
  ds(:,k+1) = x - xts(:,k+1);  % actual estimate error
  ds(:,k+1) = fix_state(ds(:,k+1));
end

subplot(1, 3, 1)

plot(xts(1,:), xts(2,:), '--g','LineWidth',3)
hold on
plot(xs(1,:), xs(2,:), '-b','LineWidth',3)
legend('true', 'estimated')

xlabel('x')
ylabel('y')
axis equal
axis xy

% beacon
plot(S.pbs(1,:), S.pbs(2,:), '*r');

for k=1:1:N
  plotcov2(xs(1:2,k), 1.96^2*Ps(1:2,1:2,k));
end
quiver(xts(1,:), xts(2,:), .5*cos(xts(3,:)), .5*sin(xts(3,:)),'g');
quiver(xs(1,:), xs(2,:), .5*cos(xs(3,:)), .5*sin(xs(3,:)),'b');

subplot(1,3,2)

plot(ds')

mean(sqrt(sum(ds.*ds, 1)))
xlabel('k')
ylabel('meters or radians')
legend('e_x','e_y','e_\theta','e_r')
%The bearing-only sensor is quite not enough to reconstruct the full state,
%as shown by non-stabilizing noise errors (e_r does not converges to zero).

subplot(1,3,3)
```

```
plot(ts, reshape(sqrt(Ps(1,1,:)),N+1,1), ...
     ts, reshape(sqrt(Ps(2,2,:)),N+1,1), ...
     ts, reshape(sqrt(Ps(3,3,:)),N+1,1));
legend('\sigma_x','\sigma_y','\sigma_\theta')

xlabel('t')
ylabel('meters or radians')


function [x, varargout] = uni_f(x, u, S)
% dynamical model of the unicycle
c = cos(x(3));
s = sin(x(3));

x = [x(1) + c*u(1)*x(4);
     x(2) + s*u(1)*x(4);
     x(3) + u(2);
     x(4)];

x = fix_state(x, S);

if nargout > 1
  % F-matrix
  varargout{1} = [1, 0, -s*u(1)*x(4), c*u(1);
                  0, 1, c*u(1)*x(4), s*u(1);
                  0, 0, 1, 0;
                  0, 0, 0, 1];
end


function [y, varargout] = br_h(x, S)

p = x(1:2);

y = [];
H = [];
for i=1:size(S.pbs, 2)
  pb = S.pbs(:, i); %i-th beacon
  d = pb - p;
  r = norm(d);

  th = fangle(atan2(d(2), d(1)) - x(3));
  y = [y; th; r];

  if nargout > 1
    % H-matrix
    H = [H;
         d(2)/r^2, -d(1)/r^2, -1;
         -d'/r, 0];
  end
end

if nargout > 1
  varargout{1} = H;
end


function [y, varargout] = b_h(x, S)

p = x(1:2);
```

```
y = [];
H = [];
for i=1:size(S.pbs, 2)
  pb = S.pbs(:, i); %i-th beacon
  d = pb - p;
  r = norm(d);

  th = fangle(atan2(d(2), d(1)) - x(3));
  y = [y; th];

  if nargout > 1
    % H-matrix
    H = [H;
         d(2)/r^2, -d(1)/r^2, -1,0];
  end
end

if nargout > 1
  varargout{1} = H;
end



function [x,P] = ekf_predict(x, P, u, S)

[x, F] = S.f(x, u, S);
x = fix_state(x, S);  % fix any [-pi,pi] issues
P = F*P*F' + S.Q;


function [x,P] = ekf_correct(x, P, z, S)

[y, H] = S.h(x, S);
P = P - P*H'*inv(H*P*H' + S.R)*H*P;
K = P*H'*inv(S.R);

e = z - y;
e = fix_meas(e, S);  % fix any [-pi,pi] issues
x = x + K*e;


function x = fix_state(x, S)
x(3) = fangle(x(3));

function z = fix_meas(z, S)
z
S.r
for i=1:size(S.pbs,2)
  if S.bearing_only
    z(i) = fangle(z(i));
  else
    z(2*i-1) = fangle(z(2*i-1));
  end
end

function a = fangle(a)
% make sure angle is between -pi and pi
a = mod(a,2*pi);
if a < -pi
  a = a + 2*pi;
else
```

```
  if a > pi
    a = a - 2*pi;
  end
end
```

```
z =

    0.3864
    0.0309


ans =

     2


z =

   -0.0477
    0.0878


ans =

     2


z =

    0.1355
    0.0505


ans =

     2


z =

   -0.3527
   -0.7778


ans =

     2


z =

   -0.1110
    0.1054


ans =

     2
```

```
z =

   -0.0936
   -0.4072


ans =

     2


z =

   -0.1386
    0.0471


ans =

     2


z =

    0.2595
   -0.1294


ans =

     2


z =

   -0.2661
    0.1760


ans =

     2


z =

    0.2150
   -0.1255


ans =

     2


z =

    0.0846
    0.0490
```

```
ans =

     2


z =

    0.1496
    0.4253


ans =

     2


z =

    0.1279
   -0.7591


ans =

     2


z =

    0.0574
   -0.6478


ans =

     2


z =

    0.1315
   -1.0240


ans =

     2


z =

   -0.1499
   -0.6143


ans =

     2
```

```
z =

    0.1674
    0.9199


ans =

     2


z =

    0.7004
    0.0241


ans =

     2


z =

   -0.1934
    0.2358


ans =

     2


z =

    0.0415
    0.2550


ans =

     2


z =

   -0.1510
    0.0663


ans =

     2


z =

    0.1051
    0.0433
```

```
ans =

     2


z =

    0.2008
   -0.2772


ans =

     2


z =

    0.2280
    0.2186


ans =

     2


z =

   -0.0319
   -0.0859


ans =

     2


z =

    0.1520
    0.1874


ans =

     2


z =

    0.0713
    0.3045


ans =

     2


z =
```

```
    0.1334
    0.1347


ans =

     2


z =

   -0.0113
   -0.0600


ans =

     2


z =

   -0.0341
    0.1126


ans =

     2


z =

   -0.1325
   -0.4088


ans =

     2


z =

   -0.1208
   -0.2649


ans =

     2


z =

    0.0101
   -0.1734


ans =
```

```
     2


z =

   -0.1648
    0.0120


ans =

     2


z =

   -0.0308
    0.3560


ans =

     2


z =

   -0.2572
   -0.0402


ans =

     2


z =

   -0.2382
   -0.1427


ans =

     2


z =

   -0.1556
    0.1409


ans =

     2


z =
```

```
    0.4325
   -0.0867


ans =

     2


z =

    0.0062
   -0.1418


ans =

     2


z =

   -0.0547
    0.4574


ans =

     2


z =

   -0.1570
   -0.0200


ans =

     2


z =

   -0.4715
   -0.0995


ans =

     2


z =

   -0.2912
    0.2432


ans =
```

```
     2


z =

   -0.1795
   -0.1826


ans =

     2


z =

    0.1577
    0.0158


ans =

     2


z =

   -1.1897
    0.1245


ans =

     2


z =

    0.3058
    0.2483


ans =

     2


z =

   -0.6759
    0.0384


ans =

     2


z =

    0.1870
```

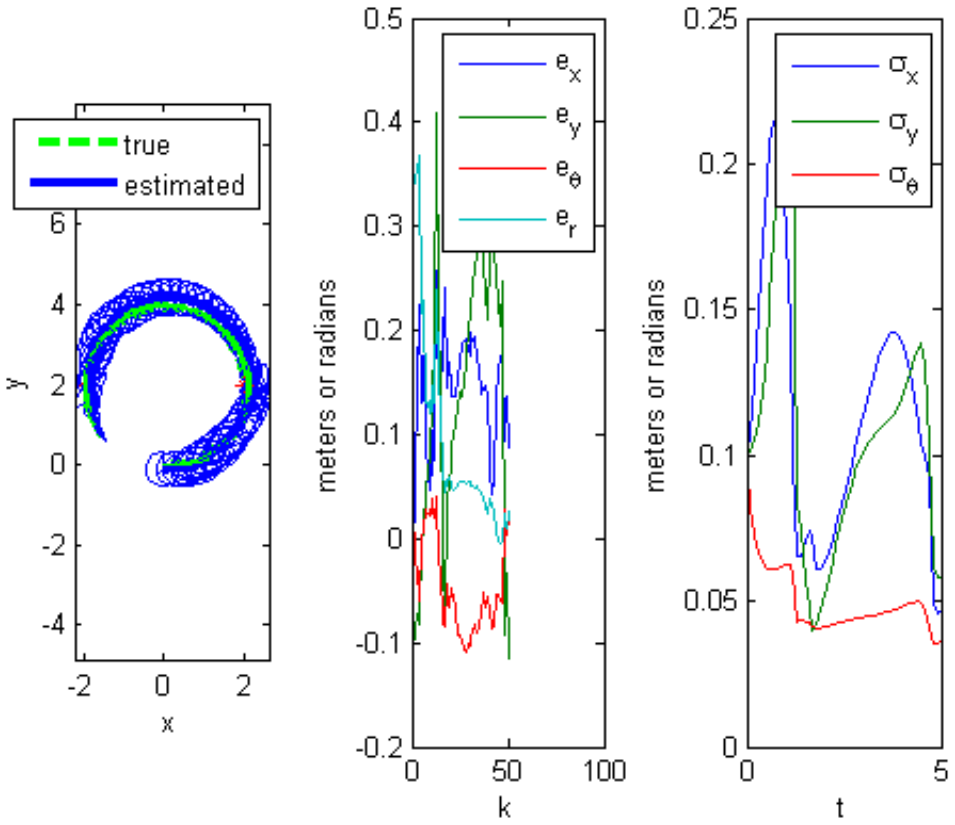
```
    -0.0289

ans =

     2

ans =

    0.2665
```



Published with MATLAB® R2014a