```
function f = uni_ekf_test
% 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 = 0;

% 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 error amount in radius stablizes to 0
%after some time. This demonstrates that r_hat becomes close to 1.04
%which is true radius plus noise covariance in the radius measurement.

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, 0;
         -d'/r, 0, 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;
         -d'/r, 0, 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
   -0.0833
   -0.1783
   -0.1835


ans =

     4


z =

    0.0525
   -0.0175
    0.3477
    0.0667


ans =

     4


z =

    0.3994
   -0.1101
   -0.3278
   -0.0290


ans =

     4


z =

   -0.1472
   -0.0637
   -0.2353
    0.0884


ans =

     4
```

```
z =

    0.8405
   -0.0500
    0.0989
    0.1241


ans =

     4


z =

   -0.1392
   -0.1267
    0.1496
    0.0630


ans =

     4


z =

   -0.3229
   -0.0079
    0.1867
    0.0470


ans =

     4


z =

   -0.0015
   -0.0223
    0.0256
    0.1757


ans =

     4


z =

    0.0453
   -0.1025
    0.0923
    0.1406


ans =
```

```
     4


z =

    0.1570
    0.0421
    0.0030
   -0.1348


ans =

     4


z =

   -0.3520
    0.0066
    0.0934
   -0.1311


ans =

     4


z =

   -0.1427
   -0.0013
   -0.1718
   -0.0829


ans =

     4


z =

    0.2008
   -0.0409
   -0.0984
   -0.0474


ans =

     4


z =

   -0.1277
   -0.0406
   -0.2250
```

```
   -0.0277


ans =

     4


z =

    0.0379
   -0.0755
    0.0306
   -0.0336


ans =

     4


z =

   -0.1145
   -0.0450
    0.2174
   -0.0682


ans =

     4


z =

    0.0538
    0.0609
    0.5159
    0.0338


ans =

     4


z =

   -0.2504
   -0.0643
    0.3034
    0.0094


ans =

     4


z =
```

```
    0.0345
   -0.0595
    0.0794
    0.0707


ans =

     4


z =

   -0.1223
    0.0195
   -0.1197
    0.0303


ans =

     4


z =

   -0.3377
    0.0404
    0.2110
    0.0222


ans =

     4


z =

    0.0542
   -0.0469
   -0.6117
    0.0711


ans =

     4


z =

    0.0106
   -0.1069
    0.0652
   -0.0185


ans =
```

```
     4


z =

    0.0338
   -0.0253
   -0.0177
   -0.1139


ans =

     4


z =

    0.0916
   -0.0688
   -0.1205
   -0.0465


ans =

     4


z =

    0.3063
   -0.0387
   -0.3324
   -0.0651


ans =

     4


z =

   -0.1947
   -0.0502
   -0.1856
    0.0011


ans =

     4


z =

   -0.1629
   -0.0243
    0.2428
    0.1206
```

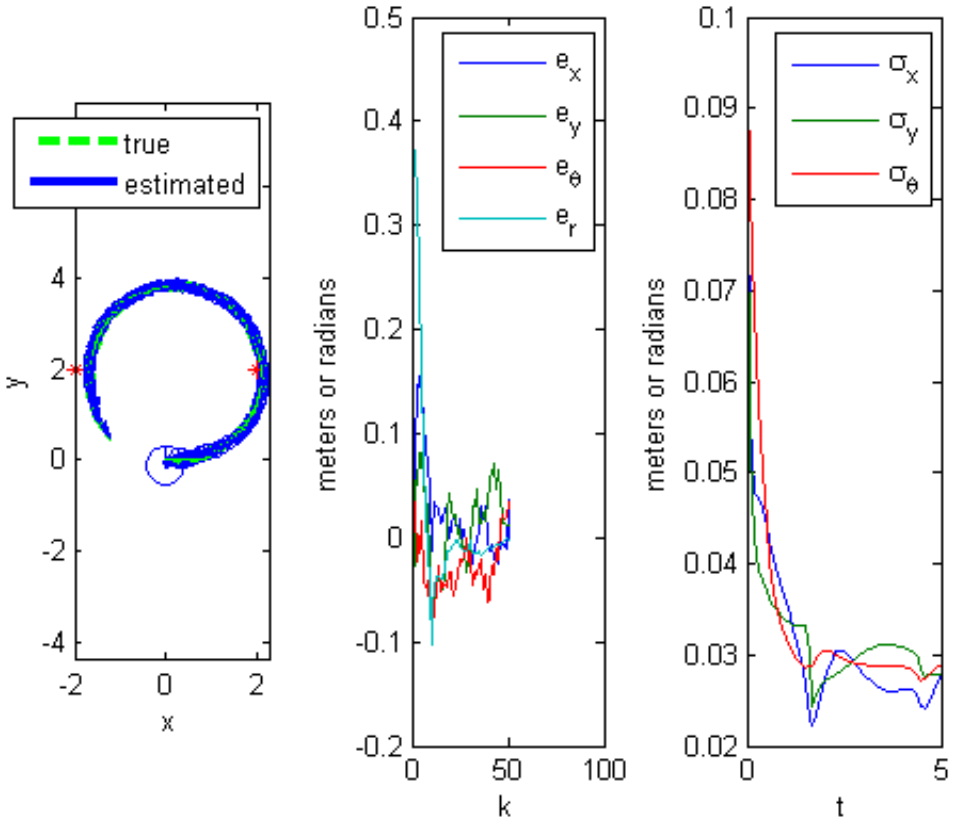
```
ans =

     4


z =

   -0.1733
   -0.0337
    0.0110
    0.1262


ans =

     4


z =

    0.0103
    0.0772
    0.0733
   -0.0786


ans =

     4


z =

    0.0352
    0.1804
   -0.2027
   -0.0460


ans =

     4


z =

   -0.1833
   -0.0099
   -0.3959
   -0.0374


ans =

     4


z =
```

```
   -0.0087
   -0.0072
   -0.2883
    0.0526


ans =

     4


z =

   -0.2667
   -0.0928
   -0.4617
   -0.0627


ans =

     4


z =

   -0.1198
    0.0193
    0.0370
   -0.1147


ans =

     4


z =

   -0.0110
   -0.0205
    0.3436
    0.1034


ans =

     4


z =

   -0.4785
   -0.0260
    0.0951
   -0.0152


ans =

     4
```

```
z =

    0.2251
    0.0095
    0.0172
    0.1336


ans =

     4


z =

   -0.1840
   -0.0424
    0.3130
   -0.0525


ans =

     4


z =

   -0.3570
   -0.0490
   -0.1023
    0.0154


ans =

     4


z =

   -0.2359
    0.0029
   -0.2207
   -0.0228


ans =

     4


z =

    0.0741
   -0.0678
    0.1481
    0.0009
```

```
ans =

     4


z =

   -0.2487
   -0.0857
   -0.1589
   -0.0854


ans =

     4


z =

   -0.0732
    0.0638
    0.2894
    0.0695


ans =

     4


z =

   -0.4108
    0.1348
   -0.3131
    0.0436


ans =

     4


z =

   -0.2132
    0.1274
    0.1080
    0.0568


ans =

     4


z =

   -0.1039
```

```
    0.0221
    0.2803
    0.0401


ans =

     4


z =

   -0.3438
   -0.0471
    0.0800
   -0.0129


ans =

     4


z =

   -0.1758
    0.0946
   -0.0901
   -0.1841


ans =

     4


z =

   -0.0440
   -0.1062
    0.1658
    0.0934


ans =

     4


ans =

    0.0886
```



Published with MATLAB® R2014a